N.B.
Coal Limited
MINTO N.B

MONITORING STRAIN ON THE TRISTRUCTURE BACKLEG KNEE PLATE OF THE MARION 8200 DRAGLINE

(PHASE I OF OPTIMIZATION OF DRAGLINE PERFORMANCE THROUGH COMPUTER CONTROL)

[Part of the Program of the Federal Panel on Energy R & D (PERD)]

Final Report
by:

Mr. Andrew Cormier
President and General Manager
N.B. Coal Limited
Minto, NB
E0E 1J0

and

Ray St-Laurent, P. Eng.
Senior Engineer
Research and Productivity Council
P.O. Box 20,000
Fredericton, NB
E3B 6C2
(Subcontractor to NB Coal Ltd.)

Work on the project was conducted under the auspices of the Canada Centre for Mineral and Energy Technology, Energy, Mines and Resources Canada.

DSS File No. 06SQ.23440-8-9092EX00PT04

DSS Contract Serial No. 23440-8-9092/01-SQ

CANMET Scientific Authority: Dr. V. Srajer

(RPC Report No. ELR/89/06)

67pp

Date: March 28, 1989

MICROMEDIA

ABSTRACT

New Brunswick Coal has completed the first phase of a four phase project to enhance performance and increase safety in the operation of a Marion 8200 dragline by application of computer monitor/control techniques. Phase 1 dealt with the installation and automatic monitoring of strain gauges in the high cyclic stress region of the tristructure backleg knee plate. This area previously required stress repairs. The monitoring system emits an audiovisual alarm for the operator if any gauge's stress exceeds its limits. The trigger limits are easily field-settable. Overstress events result in the recording of stress data for every gauge over a sixty second period starting forty seconds before the alarm commenced. The maximum and minimum stress values for each gauge are recorded hourly. One button on the monitor box graphs the stresses on its display for the latest alarm period. Another button summarizes the events recorded. The robust, easy-to-use system can differentiate between individual operators and the nature of the overburden under excavation. This monitoring system will be integrated with the systems developed in the remaining phases of the project.

RÉSUMÉ

New Brunswick Coal a terminé la première phase d'un projet à quatre phases en vue d'améliorer le rendement et la sécurité de sa pelle à benne traînante Marion 8200 au moyen de techniques de surveillance et de contrôle informatisées. La phase I s'est penchée sur l'installation et la surveillance automatique de jauges de contrainte dans la zone de la plaque coudée du chevalet arrière de la triple structure, qui subit des contraintes cycliques excessives. Dans le passé, cette zone a nécessité des réparations de dommages causés par la tension. Le système de surveillance envoie au conducteur une alarme audio-visuelle si la tension enregistrée par la jauge dépasse les limites. Il est facile de régler sur place les seuils de déclenchement. Les contraintes excessives entraînent l'enregistrement des données pertinentes sur la jauge pendant soixante secondes, dont quarante avant l'émission de l'alarme. La touche d'un bouton donne sur l'écran un graphique matriciel des contraintes survenues lors de la dernière période donnant naissance à une alarme. La touche d'un autre bouton donne un résumé des données enregistrées. Ce système, solide et convial, peut distinguer entre les différents conducteurs et cerner le type de terrains-morts en excavation. Ce système de surveillance sera intégré à d'autres systèmes qui seront mis au point lors de la réalisation des autres phases du projet.

EXECUTIVE SUMMARY

Draglines used in coal mines operate in a high cyclic stress environment with severe lubrication requirements. They offer an excellent opportunity to enhance performance and safety through use of computer monitor/control techniques. Phase I of this project dealt with the monitoring of strain on the tristructure backleg knee plate of the Marion 8200 dragline operating at Minto, New Brunswick. NB Coal subcontracted the work for this phase to the New Brunswick Research and Productivity Council (RPC). The majority of this work, and its management, was performed by the Electronics department at RPC.

This phase was especially relevant to NB Coal since stress repairs have been previously required in the backleg knee plate region of the dragline.

Strain gauges were installed at thirteen locations to measure the stress regions at and near the knee plate area. A continuously running monitoring and alert system was developed, installed and commissioned. Off the shelf hardware components were combined with custom software to monitor the stress at selected points. Overstress events, as well as hourly maximum and minimum stress values for each gauge, are recorded on removable diskettes

An audio-visual alarm alerts the operator when stress on a gauge exceeds preset limits. An alarm event results in the stress history being stored for the time 40 seconds before the alarm to 20 seconds following. Pressing a button displays the gauges' stress curves for the alarm period. Another button summarizes the alarm events recorded on the disk.

Phase I has met or exceeded the goals set out in the statement of work. Although the system has only recently been installed, it has already been eagerly accepted by NB Coal's supervisory staff as a knowledge tool. The hourly stress ranges can be used to identify the operator and the nature of the overburden under excavation.

RÉSUMÉ

Les pelles à benne traînante utilisées dans l'exploitation houillère sont normalement sujettes à des conditions de contraintes cycliques rigoureuses, ce qui nécessite une lubrification intense et fréquente. Il y a lieu d'améliorer sensiblement le rendement et la sécurité de ces équipements grâce à des techniques de surveillance et de contrôle informatisées. La phase I de ce projet s'est penchée sur la surveillance de la tension subie par la plaque coudée du chevalet arrière qui fait partie de la triple structure de la pelle à benne traînante 8200 utilisée à Minto, Nouveau-Brunswick. NB Coal a confié le contrat de cette phase au Conseil de la recherche et de la productivité (CRP) du Nouveau-Brunswick. La section Électronique du CRP a entrepris la plus grande partie du travail ainsi que la gestion afférente.

Cette phase a été particulièrement pertinente pour NB Coal, étant donné que dans le passé, la plaque du chevalet arrière de la pelle à benne traînante a nécessité des réparations à cause des tensions.

Des jauges de contrainte ont été installées à treize points de la plaque coudée ou près d'elle pour mesurer les contraintes subie par cette zone. Un système de surveillance et d'alerte en continu a été mis au point, installé et vérifié. Des dispositifs disponibles sur stock ont été combinés avec des logiciels conçus spécifiquement à cette fin pour mesurer les contraintes à des endroits ponctuels. Les niveaux des contraintes excessives, ainsi que les maximums et les minimums horaires prises par chaque jauge sont enregistrés sur des disquettes amovibles.

Une alarme audio-visuelle prévient l'opérateur si le niveau de contrainte dépasse les limites préétablies. Quand un événement donnant lieu à une alarme survient, les données relatives à la contrainte pour une période de quarante secondes avant l'alarme et vingt secondes après sont enregistrées. À la touche d'un bouton, les courbes de contrainte mesurées par les jauges pendant cette période d'alarme apparaissent. La touche d'un autre bouton donne un résumé des données d'alarme enregistrées sur la disquette.

Les objectifs fixés dans la définition du but de la phase I ont été atteints, même dépassés. Bien que le système vienne d'être installé, il est déjà accepté comme un outil de travail satisfaisant par le personnel de gestion de NB Coal. Les gammes de contraintes horaires peuvent servir à identifier le conducteur et la nature des morts-terrains en excavation.

TABLE OF CONTENTS

1.0 INTRODUCTION 1

2.0 TECHNICAL DISCUSSION 2

2.1 Hardware 4

2.2 Software 8

3.0 RESULTS, CONCLUSIONS 10

1.0 INTRODUCTION

This phase of the project was especially important to NB Coal since stress repairs have been previously required in the backleg knee plate region of the dragline.

The technical requirements will be discussed on the basis of the Phase I tasks outlined in the original proposal submitted to Supply and Services Canada, DSS RFP 06SQ.233440-8-90 92. No deviations were necessary from the task specifications.

1. Develop a system to measure and record strain values.

   Off the shelf hardware was used with custom software to achieve a versatile, reliable system that was easy to use by NB Coal personnel. The system was also designed to be readily incorporated into the overall system to be developed in subsequent phases of this project.

2. Select sites for the placement of strain gauges, install and calibrate them.

   Consultation occurred with the Metallurgy Section at RPC and the finite element analysis done at University of New Brunswick was examined. Appendix A includes a copy of the finite element analysis report. Thirteen strain gauge locations were established on and near the back knee plate. Knee plate environs were chosen to permit the source of high knee plate stresses to be located.

3. Instruct dragline operating staff about the project and obtain their assistance for the monitoring tests.

   To date, we have received every cooperation at all levels of NB Coal. NB Coal staff assisted with the electrical wiring for the strain gauge and monitoring system. They also prepared the surfaces for the strain gauge bonding and the mounting of the monitor's enclosure.

4. Operate the monitoring system. This will include adjusting of alarm thresholds.

   The system has been installed and calibrated. NB Coal supervisory staff may modify parameters after unlocking the monitor's box. They are comfortable with the system and are in the process of monitoring data and establishing criteria for setting limits.

5. Analyze information recorded from the strain monitoring during operation of the dragline. This will include both on-the-spot interpretations and examinations of recorded data.

   The "hourlies" (hourly report of stress minima and maxima for each gauge) have been used to identify operator and soil type. Alarm limits are set to detect abnormalities.

   An alarm event results in gauge data been recorded every second for a one minute period commencing 40 seconds before the alarm. This data may be plotted in high resolution graphics on the display. This indicates how the stress varied for the time period circumscribing the event.

6. Once an effective alarm system has been developed, a training system for management and individual operators will be devised.

   Appendix B contains a copy of the user's manual. A training course was setup for NB Coal personnel. After the session it was felt that the NB Coal personnel had sufficient knowledge to train other NB Coal staff.

2.0 TECHNICAL DISCUSSION

Four strain gauges were selected for continuous monitoring. Figure 1 specifies the location of the thirteen gauges, indicating the four chosen. The photographs in figure 2 show the installation area for the thirteen strain gauges.

Gauge #	Monitor #	Location
1	1	Outside of lower back leg
2	-	Knee plate between pin & front leg (845 mm from pin)
3	-	Knee plate between pin & front leg (610 mm from pin)
4	-	Knee plate between pin & front leg (540 mm from pin)
5	-	Knee plate between pin & front leg (483 mm from pin)
5	3	Knee plate between pin & front leg (419 mm from pin)
7	-	Inside lower back leg
8	2	Inside lower front leg
9	-	Bottom side of horizontal support
10	-	Outside of horizontal support
11	-	Centre of web, upper front leg
12	4	Top of flange near inner edge, upper front leg
13	-	Centre of top flange, upper front leg

FIGURE 1. STRAIN GAUGE LOCATIONS

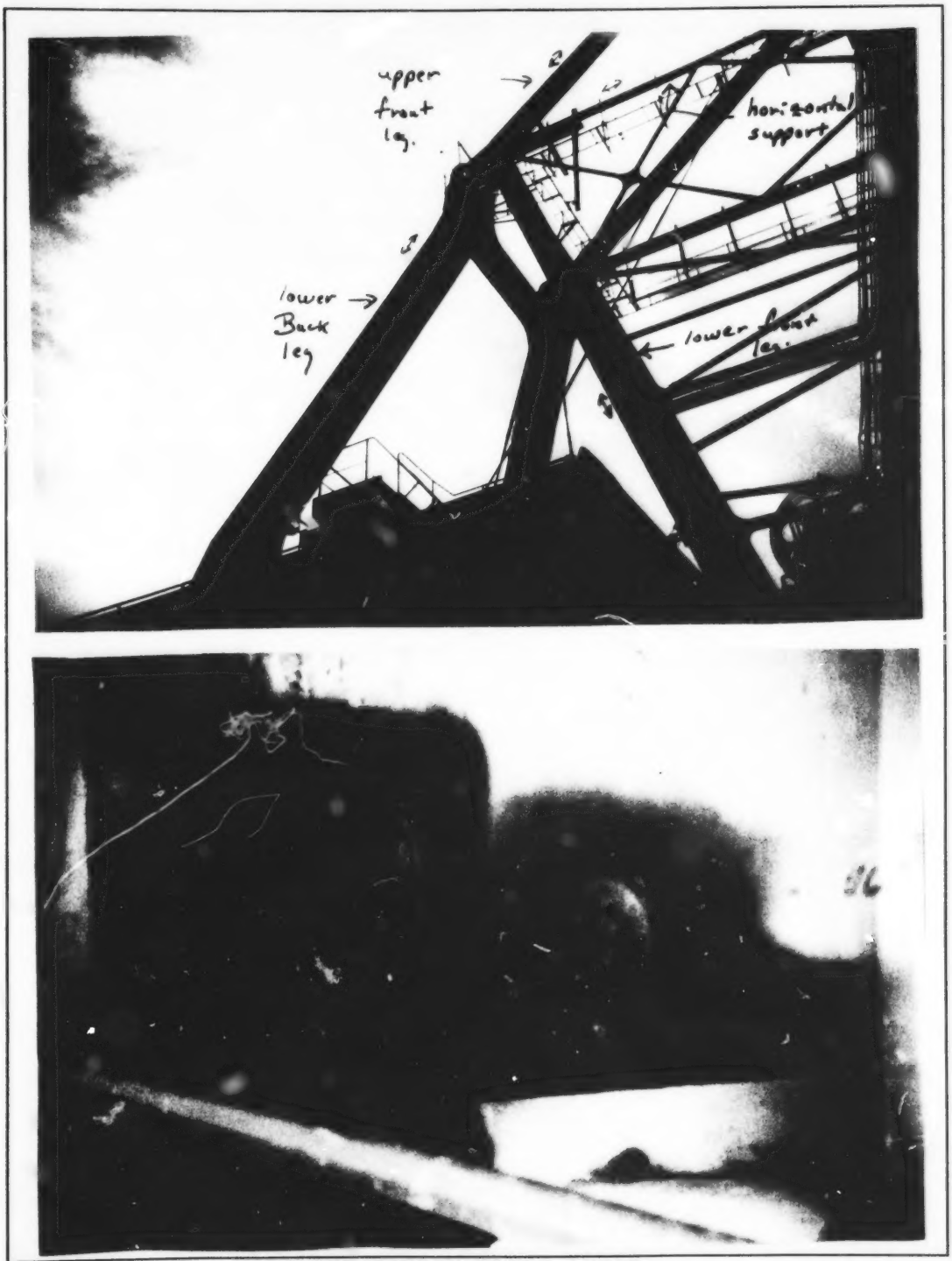


FIGURE 2. DRAGLINE STRAIN GAUGE AREAS

The installation of an appropriate monitoring system was for critical for the success of the project. It had to continuously and reliably assess, record and indicate stress behaviour. It was required to be robust, easy to maintain, simple to use, integratable with the overall system developed in subsequent phases. Both the hardware and the software had to be chosen carefully.

2.1 Hardware

The primary elements of a candidate system were:

1. Strain gauge drivers.
2. Analog-to-digital conversion.
3. Microcomputer controller.
4. Non-volatile memory.
5. User interface.
6. Expansion 'port' to connect with future system.

One candidate methodology for supplying these elements is through use of an of-the-shelf data acquisition computer. These typically combine the first four items. They tend to be robust by design. The user interface typically has limited graphics capability. The non-volatile memory can be a diskette with an uncommon format. Integrating with other systems can be very difficult. Software customisation capabilities might be limited.

The alternate approach is to base a system around an IBM PC compatible computer. Separate cards are added to handle the gauge drivers and analog to digital functions. The diskettes would be readable by any PC compatible. High resolution graphics displays are available. Available software is much greater for an industry standard such as this. A serial and parallel port are available for interconnection with future systems. Another factor was the probability that subsequent phases of the project would be based around IBM PC compatible technology.

The concerns for a PC based system are to find a model that is robust enough for the current application. Industrial rated PC's could meet this requirement. They can be bulkier then other types.

We chose to use a system based on a laptop computer. Laptops must be more rugged than desktop PC's. The integral flat screen graphics display was a factor. The use of their 3.5 inch diskettes was an advantage over the more vulnerable, and less convenient, 5.25 inch diskettes.

We selected the Toshiba T1200FB for two primary reasons:

1. It can take the analog to digital expansion card.

2. It automatically switches to its internal battery power if there is an AC power outage. Most others will go into a system restart mode if there is a power glitch. It was paramount to protect the data. If a glitch occurred during a disk write, we did not want to corrupt the file.

Quality cards, from reputable sources, comprised the rest of the major system elements. Figure 3 summarises the hardware components used. Figure 4 outlines the inputs and outputs for the AIO8 interface board.

These components, associated wiring and the power supply are all housed in a Hammond wall mounted enclosure with locking handle. The enclosure has a window through which the T1200 display is visible. Two buttons are on the box. One plots the stresses for the last alarm event. The other displays a summary of events recorded. The photograph in figure 5 shows the interior of the enclosure. Appendix D contains a copy of the wiring diagram for the dragline monitor.

The enclosure is mounted in the auxiliary operator's cab. An alarm horn and light housing is in the operator's cab. It contains a button to test the system, including the horn and light.

ITEM	#	PURPOSE
T1200FB	1	Microcomputer, user interface
WonUnder board	1	Adapts T1200 to take IBM card
AIO8-B interface board	1	Analog to digital conversion. Also accepts inputs from push-buttons and controls alarm.
AC1213 with 2B31 conditioner	4	Strain gauge drivers

FIGURE 3. MAJOR HARDWARE ELEMENTS OF MONITOR SYSTEM

ITEM	USAGE
Analog Input 1	Strain gauge #1
Analog input 2	Strain gauge #1
Analog Input 3	Strain gauge #1
Analog Input 4	Strain gauge #1
Analog Input 5	Gain calibration level
Analog Input 6	Zero calibration level
Analog Input 7	Zero calibration level
Analog Input 8	AC powerline level
Digital Input 1	Test button
Digital Input 2	List event "summary" button
Digital Input 3	Display stress plots of "Last Event" button
Digital Output 1	Alarm horn
Digital Output 2	Alarm light
Digital Output 3	Unused

FIGURE 4. AIO8 INTERFACE BOARD INPUTS AND OUTPUTS

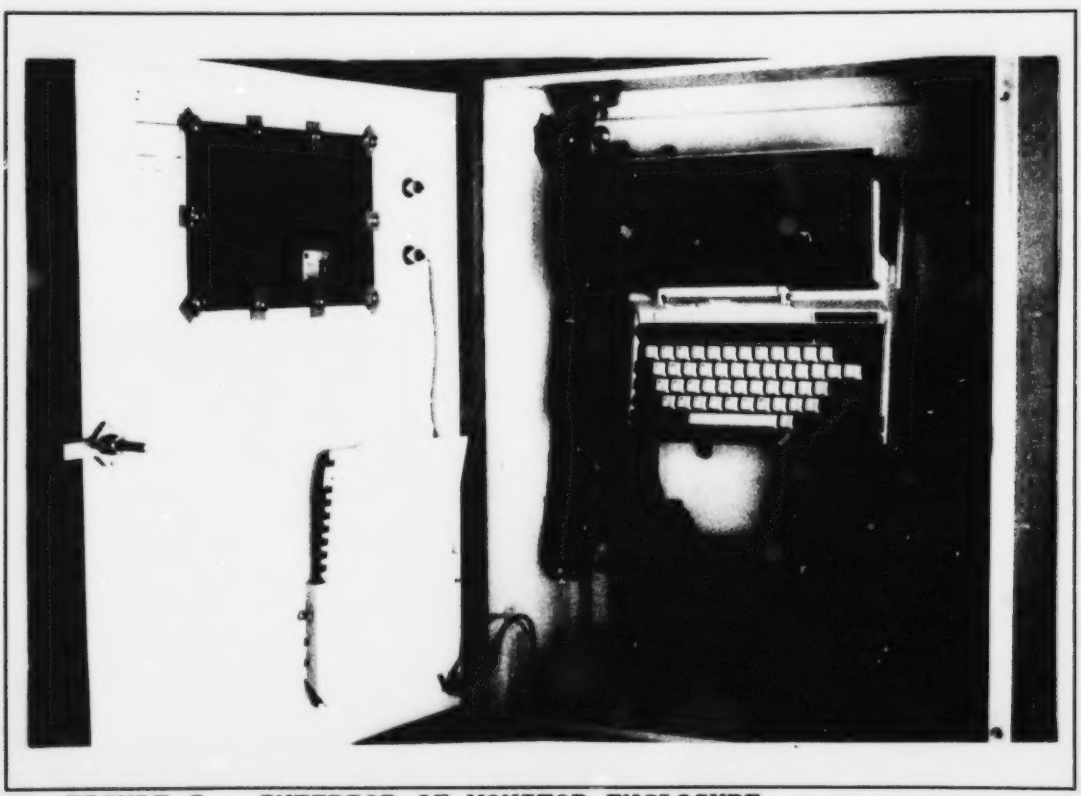


FIGURE 5. INTERIOR OF MONITOR ENCLOSURE

2.2 Software

Software requirements can be summarised as follows:

1. Easy to use with minimal instructions.
2. Monitor each strain gauge.
3. Allow individual, field-settable trigger limits for each gauge.
4. Store event data records on diskette. This is 60 sets of gauge stresses, at one second intervals, beginning 40 seconds before the overstress.
5. Every hour store a routine data record on diskette. This is the maximum and minimum absolute stresses encountered for each gauge over the preceding hour.
6. When an alarm occurs, activate an alarm horn and light in the operator's cab.
7. Allow the press of a button to display the plots of stresses for the last alarm.
8. Allow the press of a button to list an event summary.
9. Allow a button on the alarm box to test the system, including horn and alarm light.
10. Permit future interconnection to an overall system control computer which may contain expert system capabilities.

No off-the-shelf software was found that was capable of servicing the above needs. Custom software was written using the language FORTH. It has a number of features that made it an appropriate choice:

1. It is designed for equipment monitoring and control.
2. Its efficient interpreter/compiler nature simplifies interfaces and debugging.
3. It has been used for large networked systems.
4. It has built-in multi-tasker multi-user capabilities.
5. It has built-in graphics commands.
6. It has a built-in turnkey system producer. This enables the software to self-start when power is turned on.
7. The RPC team members had relevant recent experience in polyFORTH.

The software was developed under the standard MS-DOS operating system. This enable the diskfiles created to be readable by other MS-DOS systems.

Multi-tasking was used to separate data acquisition from interpretation and control. The task "REPORTER" services the strain gauges. It continuously reads the gauge values and stores them in a circular buffer. All other functions are handled by the second task, "OPERATOR". It performs alarm identification, disk access, alarm activation, control of the display, and the human interface. Additional features are outlined below:

1. Data disk replacement procedure consists of touching the space bar and then following the directions on the screen to know when to change disks.

2. The running display for the monitor program shows the date and time the current disk was installed, number of events, and percentage of disk space remaining. A small flashing face at the top of the screen signifies the monitor program is functioning.

3. An alarm message indicates if an un-acknowledged alarm exists. The alarm is acknowledged by pressing the button "Last Event". This also plots the stress curves over the 60 second event period.

4. A new line is written to the screen whenever an event or routine hourly record is written to the disk.

5. The AC power level is monitored. If a power outage is detected, any pending data is written to the disk. Then a message is displayed and the program terminates. This ensures that data will be protected from potential corruption due to a disk write with depleted batteries.

6. Calibration levels are monitored to check for unacceptable drift. A message is displayed if this occurs as this could indicate that the strain gauge readings may be inaccurate.

7. Advanced features are reached by pressing a key. The running OPERATOR program is temporarily exited. A typical reason would be to change the strain gauge alarm limits. However the REPORTER program continues to fill the circular buffer with strain gauge values. The user may take over five minutes before re-activating the OPERATOR running program without losing any data. The monitor program will catch up on the unprocessed data.

8. The alarm limits are stored in a file not on the data disk. This allows trigger limits to be changed without causing a re-compilation of the program. This file is read when the program is started.

9. When the "host" computer is installed, it will be able to change the strain gauge alarm points by communicating with the serial port in the strain monitoring system computer.

The User's Manual in Appendix B covers system operation in more detail. Appendix C contains the program listing.

3.0 RESULTS, CONCLUSIONS

The strain gauge monitoring system has been installed and is actively used by NB Coal personnel. Supervisory staff have used it already to determine operator differences and the nature of the overburden being excavated. NB Coal looks forward to the successful execution of subsequent phases of the project:

Phase 2 sets up monitoring and control of the lubrication system. This phase is scheduled for completion in September 1989.

Phase 3 will interconnect phase 1 and 2 systems to a "host computer". A data acquisition system will be established for production and safety parameters. This phase is scheduled for completion in September 1990.

Phase 4 sets up a monitoring and training system to increase the productivity of draglines. Expert system techniques will use the database established in phase 3 to provide feedback to the operator. This phase is scheduled for completion in September 1991.

APPENDICES

Appendix A. Finite Element Analysis

Appendix A. User Manual

Appendix B. Program Listing

Appendix C. Monitor Wiring Diagram

Appendix A. Finite Element Analysis

SUMMARY OF FINDINGS:

FINITE ELEMENT SIMULATIONS OF DRAGLINE DOUBLE HOIST-ROPE FAILURE AND PLATE STRESSES

for

Mr. Andy Cormier
President
N.B. Coal
Minto, N.B.

by

Robert J. Rogers, Ph.D., P. Eng.
c/o The Manufacturing Technology Centre
Department of Mechanical Engineering
University of New Brunswick
Fredericton, N.B. E3B 5A3

July 1, 1987

1.0 Introduction

The Marion Model 8200 Walking Dragline experienced a failure of both hoist ropes on October 29, 1985. An inspection of the tristructure on July 19, 1986 found that the left-hand back-leg knee plate was cracked.

The purpose of the study reported here was to simulate the dynamic response of the dragline during and following the rope failures and to determine the peak tensile stresses in the knee plate. The interpretation of the results with regard to determining a cause and effect relationship between the rope failures and the cracked plate will be left to the readers of this report and the reports by others on the subject. This report briefly describes the simulations carried out and the results. A more detailed report will be provided upon request.

2.0 Finite Element Model of Dragline

A sketch of the dragline machine and its overall dimensions is shown in Figure 1. The boom angle of 36° was obtained from the listed specifications. A commercial version of the ANSYS general purpose finite element code was used for this work. The complete model of the dragline boom, cables, tristructure and back legs is shown in Figure 2; both node and element numbers are shown.

The boom itself was modelled using six solid isoparametric elements. The density of the boom elements was adjusted so that the weight of the foot, centre-section and point matched the actual values. The modulus of elasticity of each element was adjusted so that the bending stiffness (given by EI) matched estimated average values for the corresponding portions of the boom.

The cables were modelled using 16 tension-only spar elements. Their reported densities and equivalent moduli of elasticity values were used. The two pairs of upper cables were modelled as just two cables by using twice their cross-sectional area.

The tristructure and back legs were modelled using 3-d beam elements. All the grounded connections were permitted to rotate, as were the connections between the back legs and the tristructure.

The bucket load was applied vertically downward at the nodes at the point of the boom. As well gravity loads were applied to all elements. The sag of the cables and their tension only characteristic required

the use of the ANSYS nonlinear dynamic analysis capability with stress stiffening. Due to symmetry the corresponding degrees of freedom on each side were coupled to give the same response. As well, the x and y degrees of freedom at nodes 2,3,31,62,71,72,73,81,82 and 83 were designated as master degrees of freedom. This made the problem manageable on the VAX11/750 computer.

The final simulation run assumed a structural damping factor $\beta = \zeta/(\pi f_n)$. A value for $\zeta = 0.03$ was used; suspension bridges, for example, typically have $\zeta = 0.01$ to 0.03. A fundamental natural frequency $f_n = 1.1$ cycles/sec was used based on the results of the first simulation run. Local oscillations at higher frequencies such as the cables have higher equivalent damping, in proportion to their frequency.

Two dynamic load cases were used. The first case was the "sudden" loss of the entire bucket load of 331.5 kips (1 kip = 1000 lb). This simultaneous failure of both ropes is unlikely, but provided a base case of results. The more probable sequence of failure is as follows:

(1) One cable broke suddenly relieving half the load on the boom point.

(2) The tension in the remaining cable increased as the bucket fell.

(3) The bucket reached its lowest point, with the highest tension in the cable equalling three times its original half-load tension.

(4) The remaining cable suddenly broke, relieving a load of 1.5 times the original static load on the boom.

The time duration of these events can be easily calculated if the motion of the boom is small compared to the motion of the bucket (which is reasonable). A hoist cable length of 450 ft. was used with a loaded bucket weight of 331.5 kips to give a natural period of oscillation of 1.66 seconds. A bucket fall lasting 0.8 seconds was therefore assumed. Figure 3 shows a graph of the actual loading history used for the "gradual" rope failure cases.

3.0 Dragline Simulation Results

A summary of the four dynamic load cases is shown in Table 1. As well, a number of preliminary cases, mostly static, were used to validate the model. The

Table 1 - Summary of Simulation Cases

Case	Loading	Duration	Min. time step	Damping β	Run time	Approx. Cost
1	Sudden	2 sec	0.010 sec	0	1.5 hrs	$54.
2	Gradual	5 sec	0.008 sec	0	5 hrs	$180.
3	Gradual	5 sec	0.004 sec	0	10 hrs	$360.
4	Gradual	4 sec	0.002 sec	0.0086	11.5 hrs	$414.

cost figures are for just using ANSYS and do not include other computer charges. The budget for ANSYS usage of $1540. limited the number of cases considered and precluded the running of any non-symmetric cases. Figure sets 4,5,6 and 7 show typical plots of results for the four cases.

3.1 Case 1 (Fig.4)

The sudden rope failures resulted in oscillations from and back to the starting locations. The stresses in the three members connected to the knee plate did not exceed their initial values. Between oscillations the cables went slack and so reversed stress loads did not occur. The large time step of 0.010 sec caused the cable motions to go numerically unstable after 1.5 sec.

3.2 Case 2 (Fig. 5)

The second case results show a somewhat similar pattern. The knee plate member stresses follow the gradual loading history, with overload levels reaching approximately 1.7 times the starting load. The results go unstable after about 3 seconds.

3.3 Case 3 (Fig. 6)

The third case was the same as Case 2 except that the minimum time-step was halved. The results are similar but show more high frequency oscillations between the major boom motions. Numerical instability again occurred near 3 seconds.

3.4 Case 4 (Fig. 7)

The final case used the smallest minimum time step and included non-zero damping. The results are similar with the peak stresses gradually decreasing with each bounce. Numerical instability occurs after about 3.5 sec as an accurate solution requires an even smaller time step after about 2.5 sec.

The peak loads in the tristructure and back legs occurred near 0.76 seconds. The loads are almost entirely axial. Table 2 shows the axial loads in the three knee plate members under static loading and at 0.76 seconds. The peak dynamic loads were about 65% higher than the static loads.

Member	Elem.no.	Static load (kips)	Dynamic load (kips)	% Increase
Top (approx.horiz.)	23	919.	1535.	67%
Mid (approx.vert.)	29	-616.	-1088.	63%
Back (back leg)	36	1336.	2218.	66%

Table 2 - Axial Knee Plate Member Loads

4.0 Finite Element Model of Knee Plate

The plates joining the rear members of the tri-structure and the back legs are 11 ½ inches thick steel plate. The tristructure members (numbered 23 and 29) are welded to the plate. The back-leg pivots in a 13 inch diameter hole. Another hole, 9 inches in diameter, is used to support a safety cable which has been ignored in this study. The thickness of the plate near the holes is slightly larger than elsewhere but this has been neglected. For simplicity the axial loads of the members are assumed to act in the plane of the plate.

The finite element model of the plate is shown in Figure 8. The model consists of 148 four-node quadrilateral elements. The plate is assumed to be rigidly restrained where joined to the "top" tristructure member. The applied loads are also shown in Figure 8. The four elements with the largest tensile stresses are marked with x's.

The three nodal loads for the "mid" tristructure member were obtained by distributing the axial load and the bending moment about the z-axis. The nodal loads on the back leg pivot hole were obtained by assuming a half-sine pressure distribution.

5.0 Knee Plate Stress Results

Static stress analyses were run for both the static loading and the peak dynamic loading. Stress contour plots for both cases are shown in Figure 9. The total x-direction reaction forces were within 2% of the expected loads shown in Table 2.

The tensile stresses are maximum on either side of the pivot hole as expected. The stresses rapidly decrease within the first two elements, moving radially outward. The largest principal tensile stress occurred at the bottom of the pivot hole: 24,900 psi (dynamic) and 14,900 psi (static). The largest tensile stresses near the top surface were 20,600 psi (dynamic) and 12,400 psi (static). The peak dynamic and static stresses are in the ratio 1.66 to 1. The stress contour pattern is virtually the same for the two cases. If a finer mesh had been used near the hole, the computed peak stress could be somewhat higher than observed. Time restrictions prevented the testing of finer meshes.

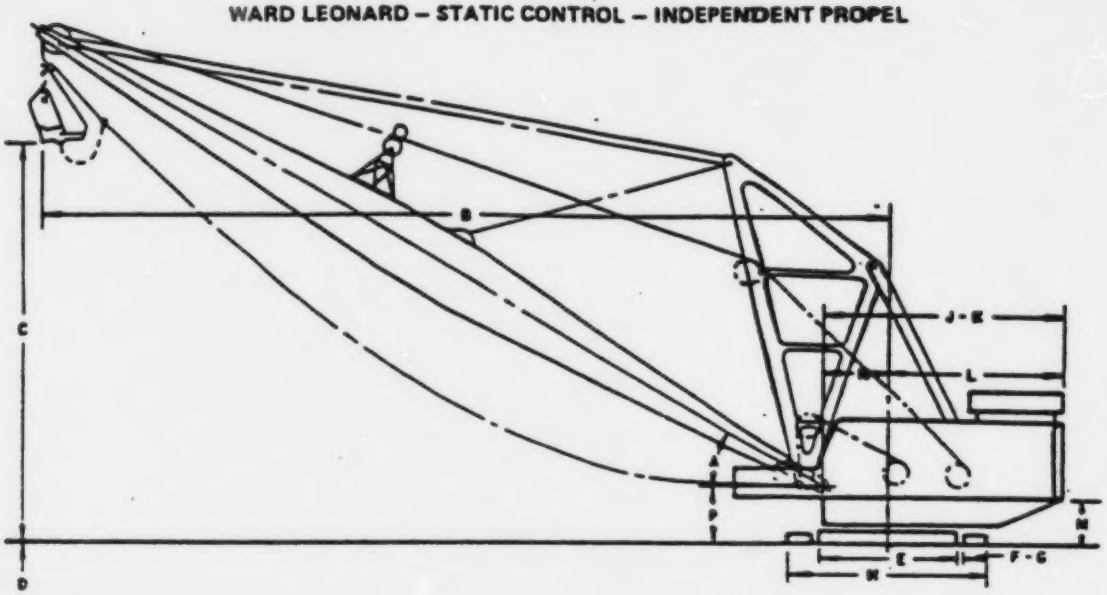
6.0 Conclusions

The dynamic response of the dragline following the gradual failure (during 0.8 sec) of both hoist ropes has been simulated. The results show that the peak loads in the members supporting the knee plate reached values 1.65 times the static load. (This compares with the sudden failure of one or both ropes where no significant overloads occur.) The peak loads occurred near 0.76 sec after the first rope failed.

The tensile stresses in the knee plate were found to be 1.66 times the full-load static values. The peak tensile stresses occurred on either side of the back leg pivot hole as expected. The peak tensile stress between the two holes was estimated to be 20,600 psi. This was the average, first principal stress across an element which was adjacent to the hole and which was about 2 inches across.

8200 WALKING DRAGLINE

WARD LEONARD – STATIC CONTROL – INDEPENDENT PROPEL



SPECIFICATION 8200-21R

Boom Length (Centerline)	350'-0"
A – Boom Angle (Approx.)	36°
B – Dump Radius	310'-0"
C – Dump Height	140'-0"
D – Depth	145'-0"
E – Tub Nominal Outside Diameter	65'-0"
Rail Circle Mean Diameter	46'-6"
Main Swing Gear Pitch Diameter	39'-9"
F – Width of Shoe	13'-0"
G – Length of Shoe	65'-0"
H – Width over Both Shoes	94'-0"
Length of Step (Approx.)	6'-0"
J – Rear End Width	67'-4"
K – Length of Rotating Frame	94'-6"
L – Rear End Clearance Radius	71'-6"
M – Clearance Under Rotating Frame	11'-0"
N – Center Rotation to Boom Foot	21'-6"
P – Ground to Boom Foot	15'-1"
Working Weight, Lbs.	8,800,000
Ballast, Lbs.	1,200,000
Maximum Allowable Load, Lbs.	355,000

Figure 1 - Marion Model 8200 Dragline

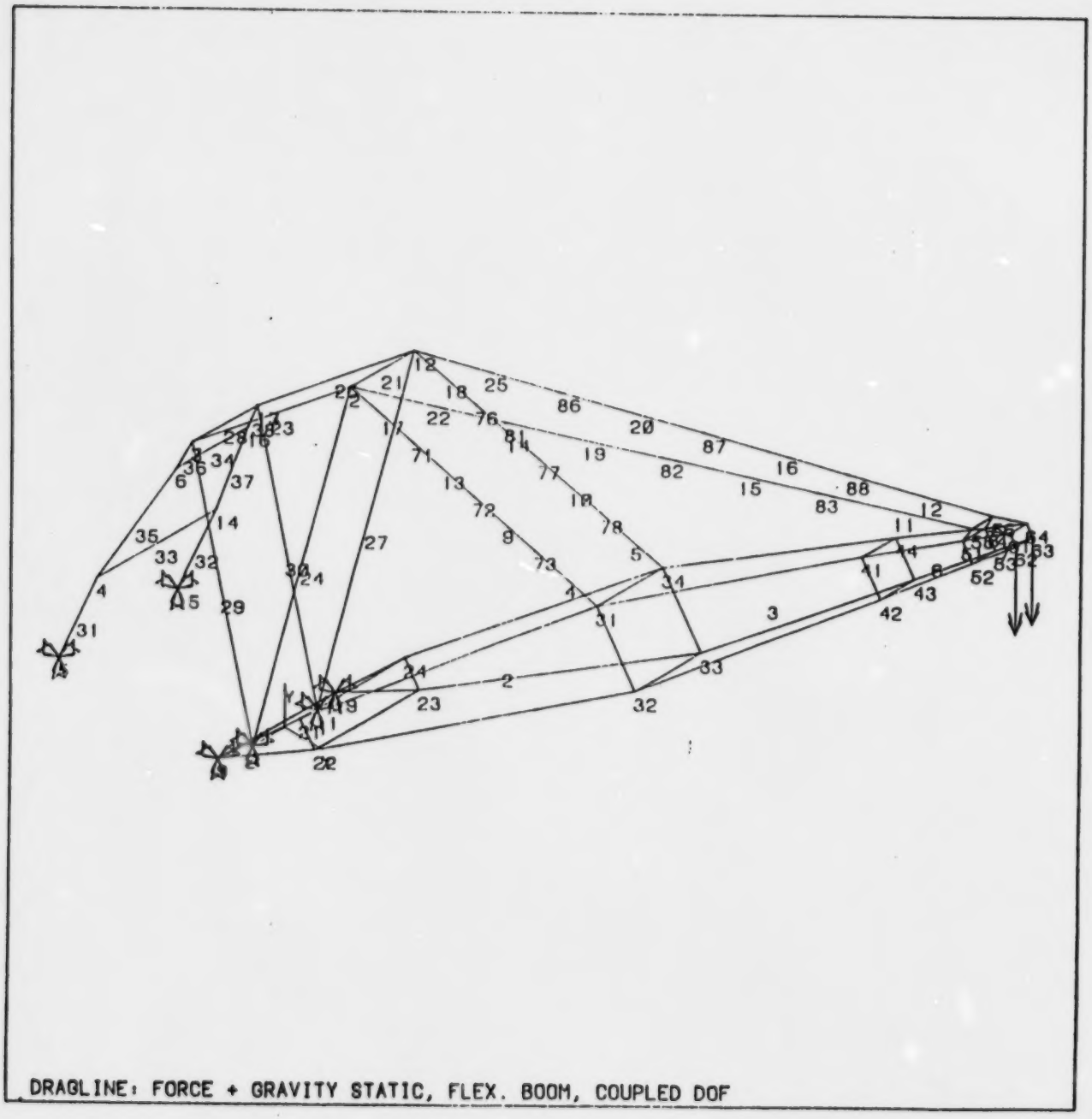


Figure 2 - Finite Element Model of Dragline

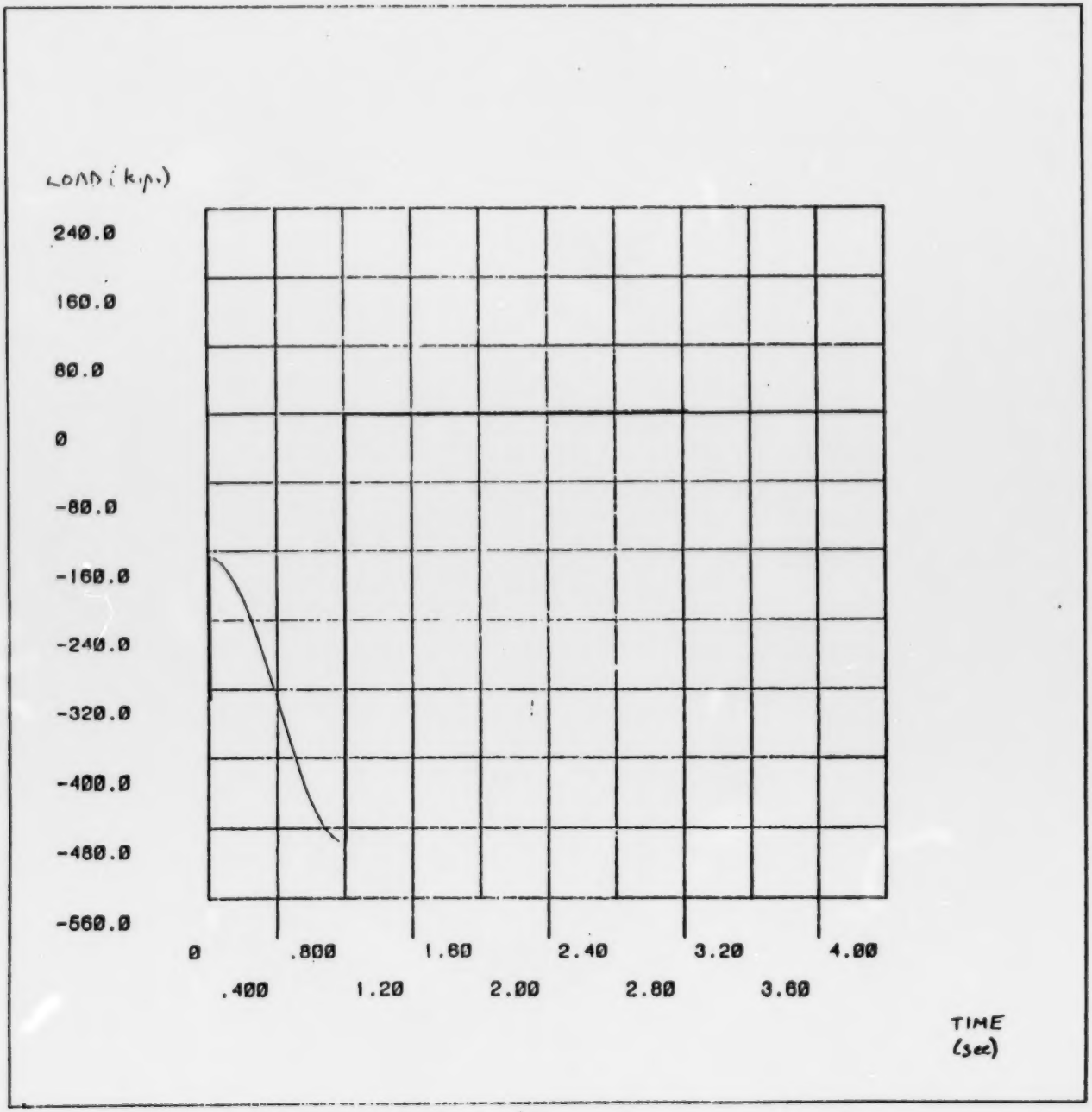


Figure 3 - Gradual Failure loading History

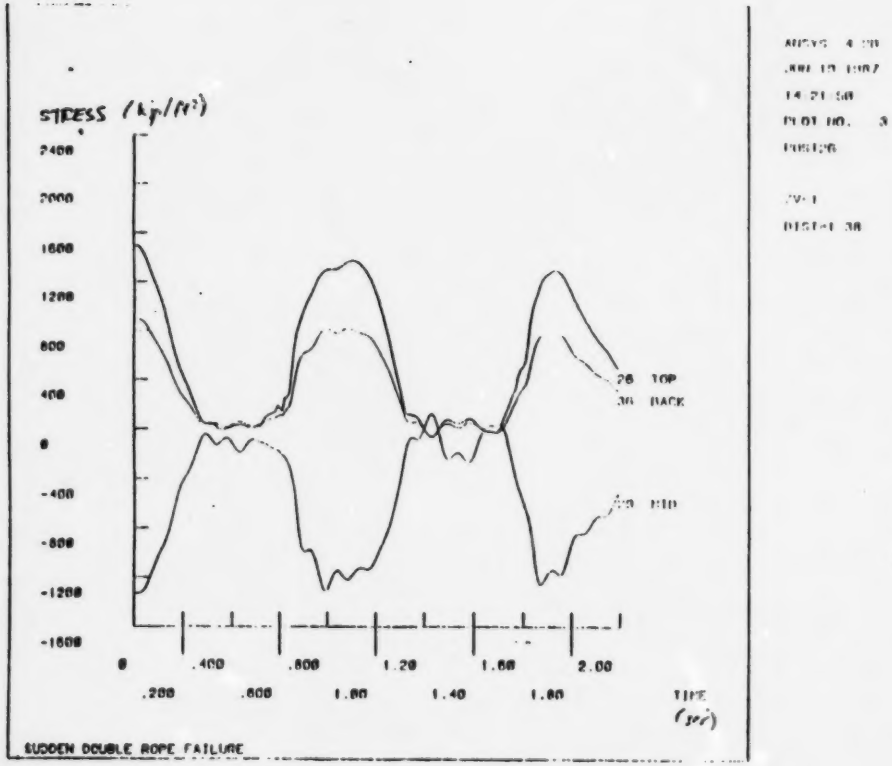


Figure 4(a) - Case 1: Knee Plate Member Axial Stresses (kips/ft^2)

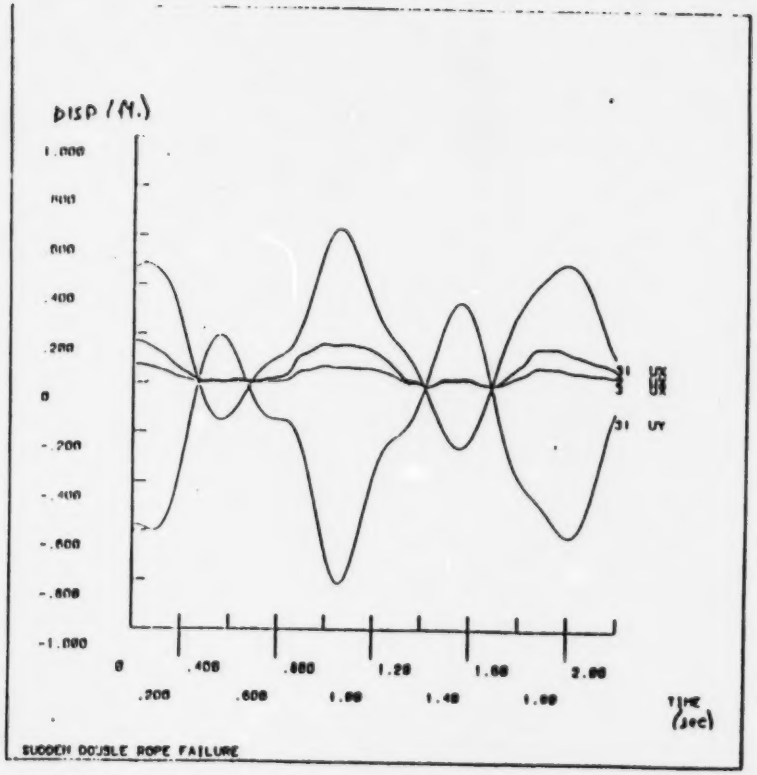


Figure 4(b) - Case 1: Mid-Boom and Structure Displacements

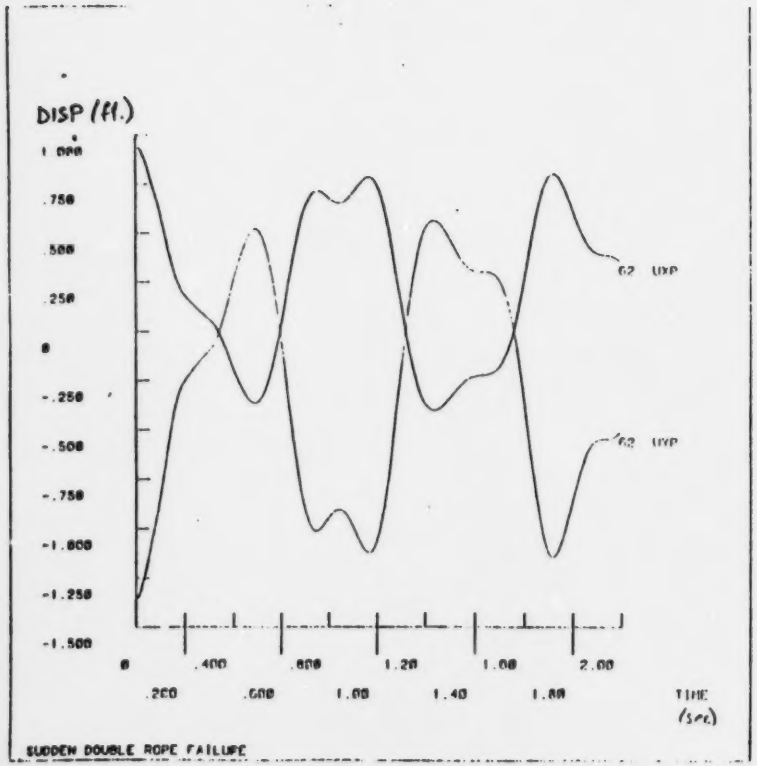


Figure 4(c) - Case 1: Boom End Displacements

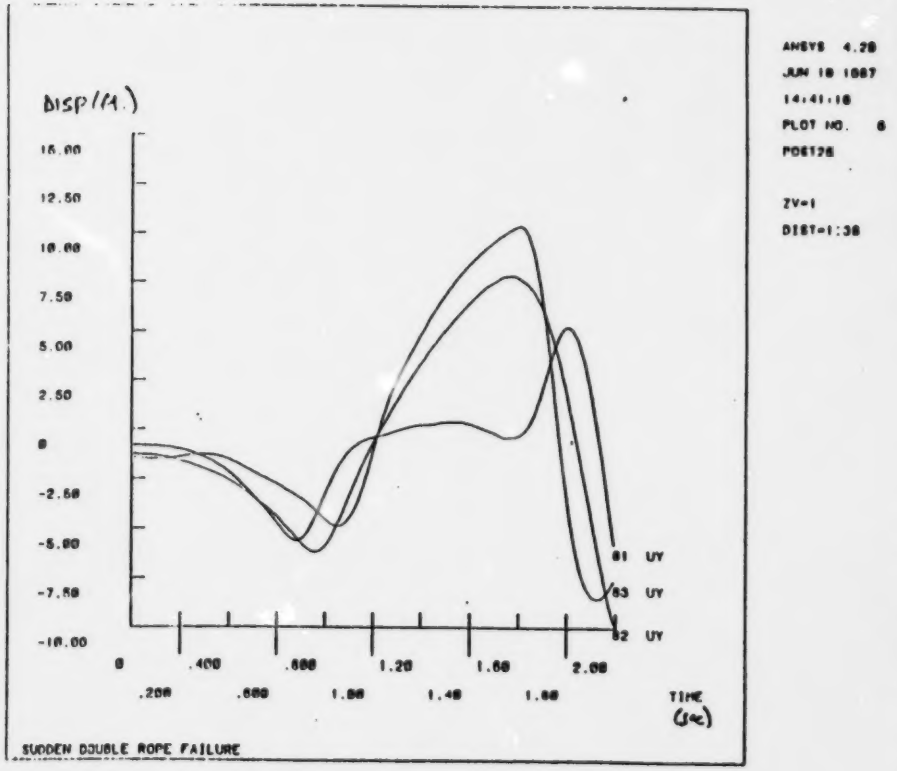


Figure 4(d) - Case 1: Upper Cable Vertical Displacements

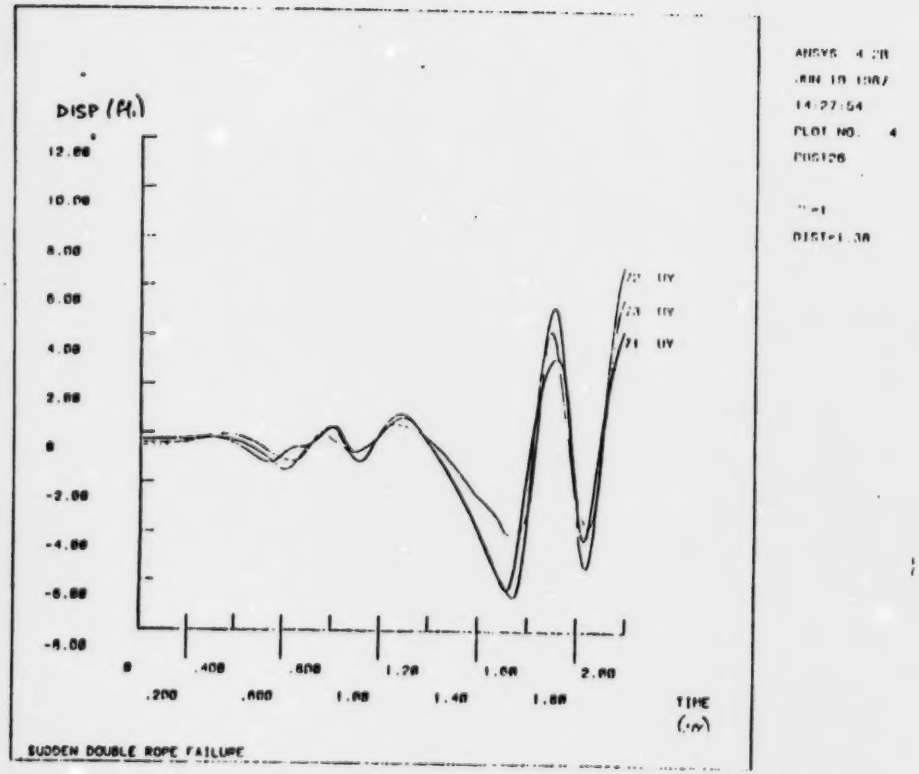


Figure 4(e) - Case 1: Mid Cable Vert. Displ.

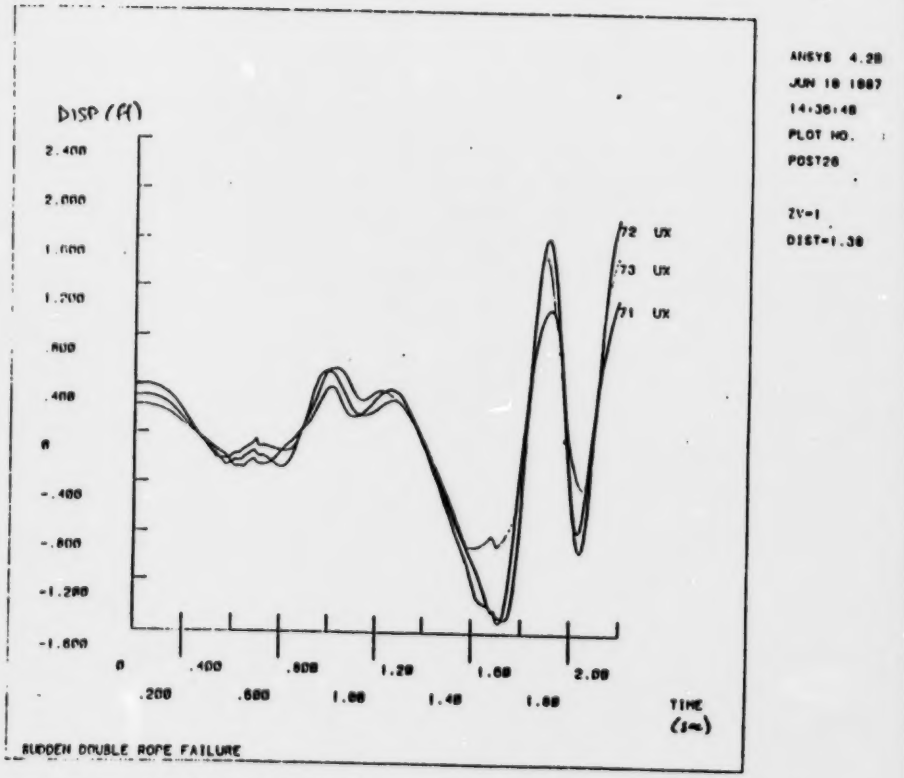


Figure 4(f) - Case 1: Mid Cable Horiz. Displ.

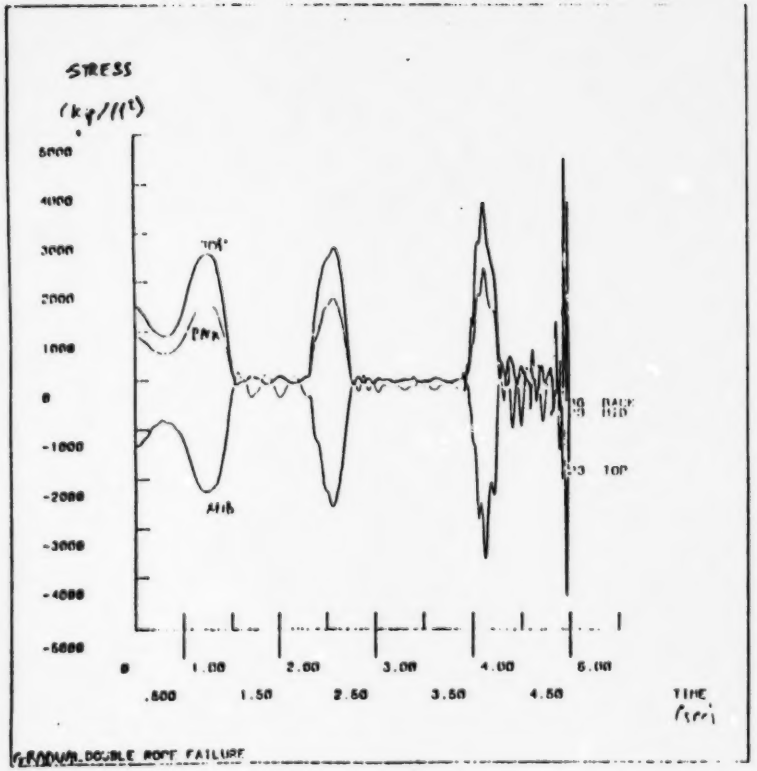


Figure 5(a) - Case 2: Knee Plate Member Axial Stresses (kips/ft²)

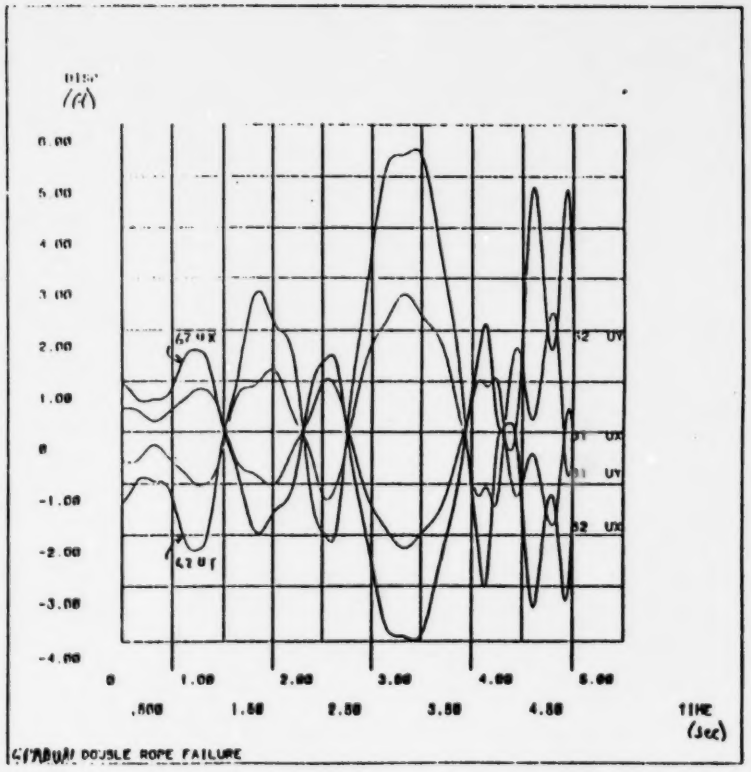


Figure 5(b) - Case 2: Boom Displacements

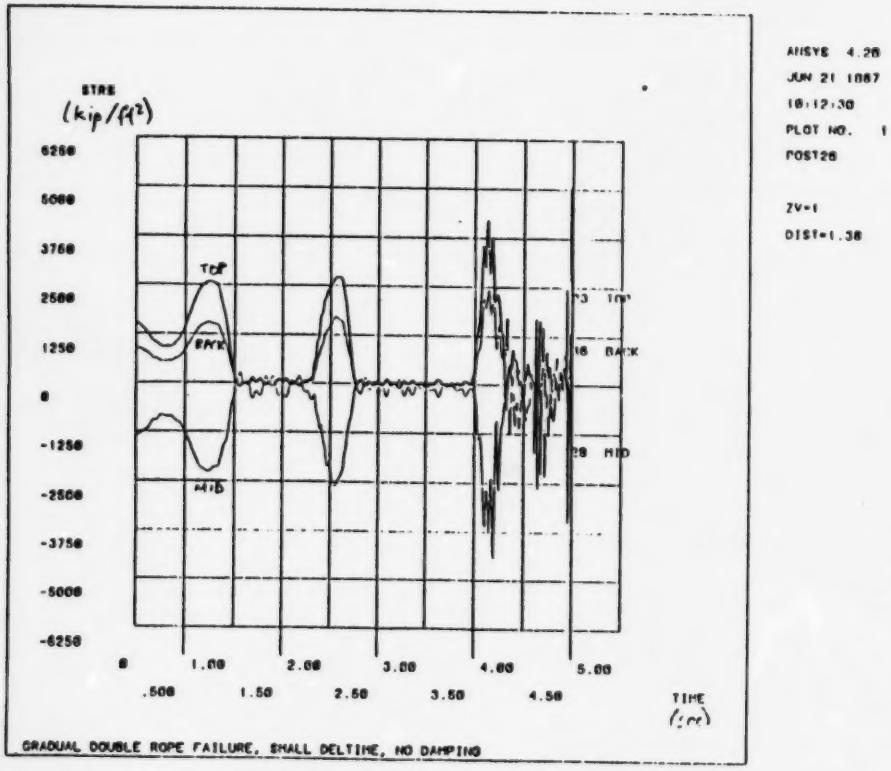


Figure 6(a) - Case 3: Knee Plate Member Axial Stresses (kips/ft^2)

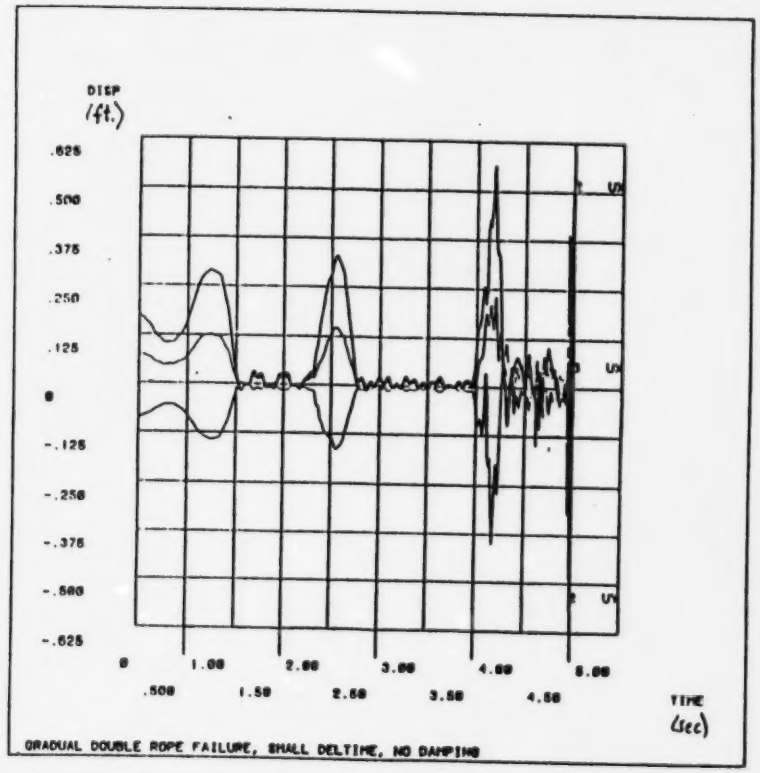


Figure 6(b) - Case 3: Tristructure Nodal Displ.

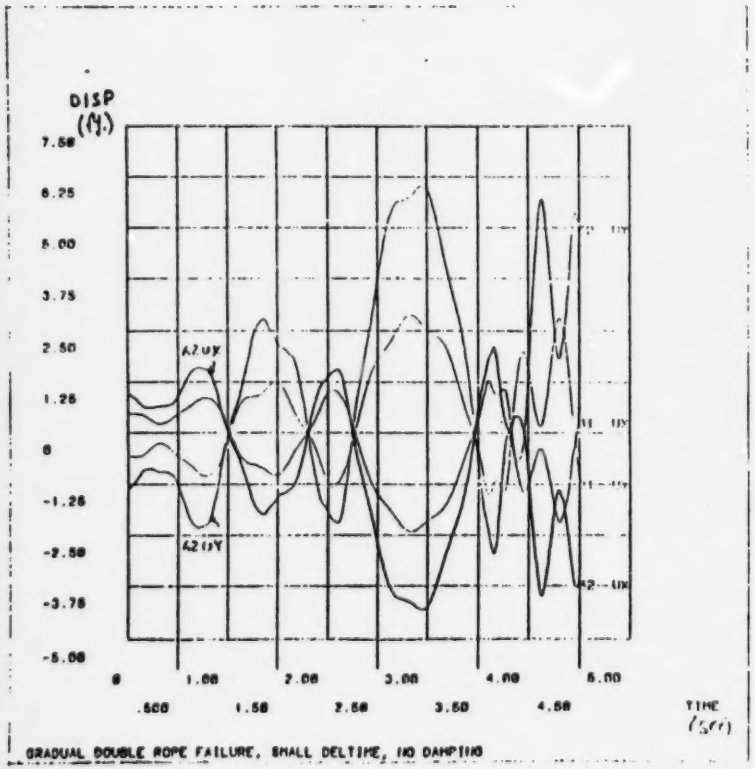


Figure 6(c) - Case 3: Boom Displ.

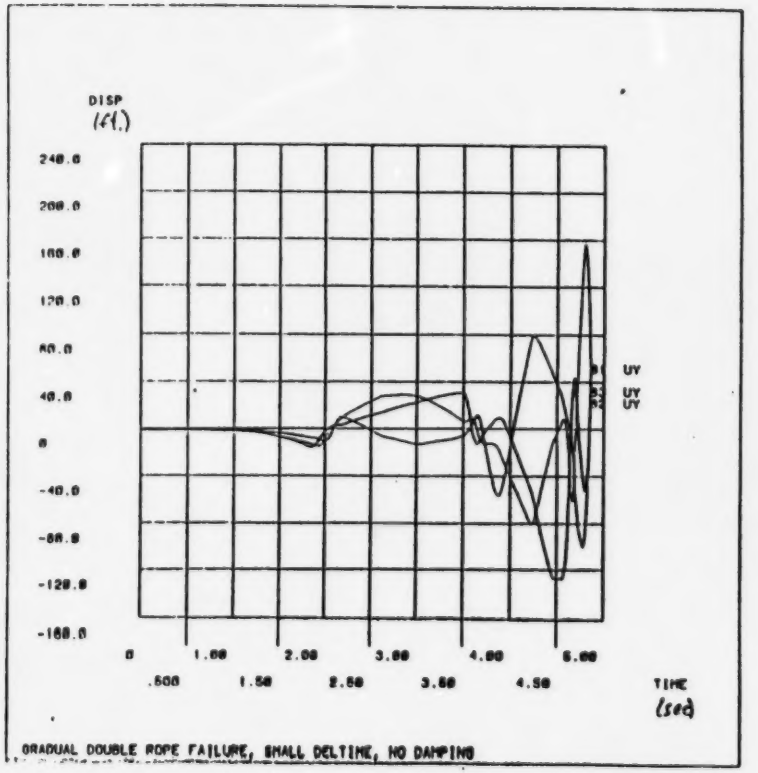


Figure 6(d) - Case 3: Upper Cable Displ.

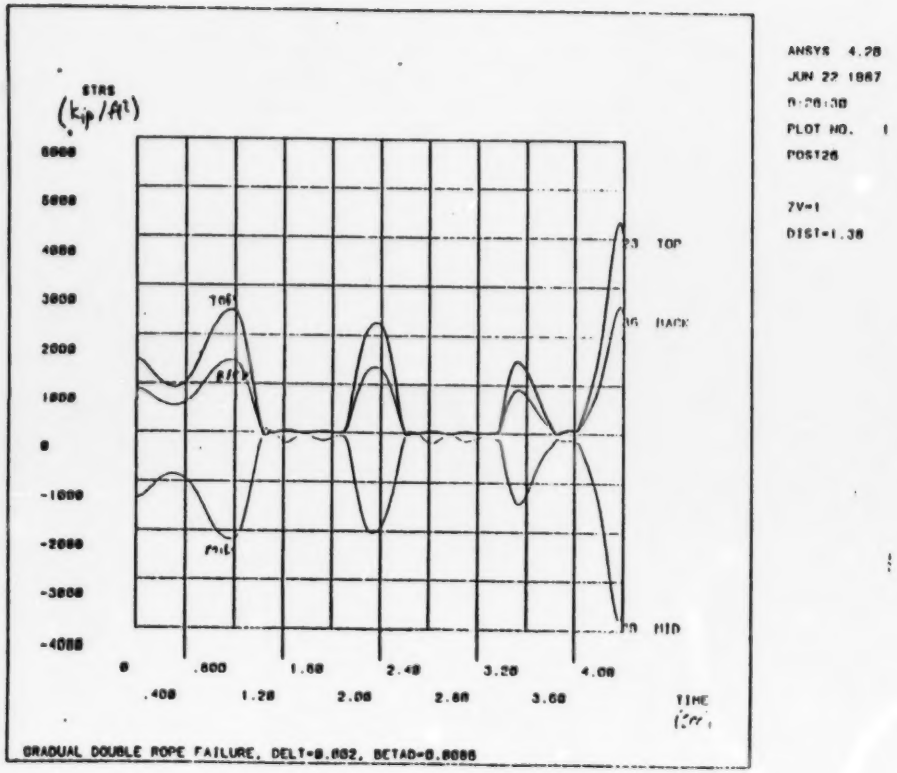


Figure 7(a) - Case 4: Knee Plate Member Axial Stresses (kip/ft^2)

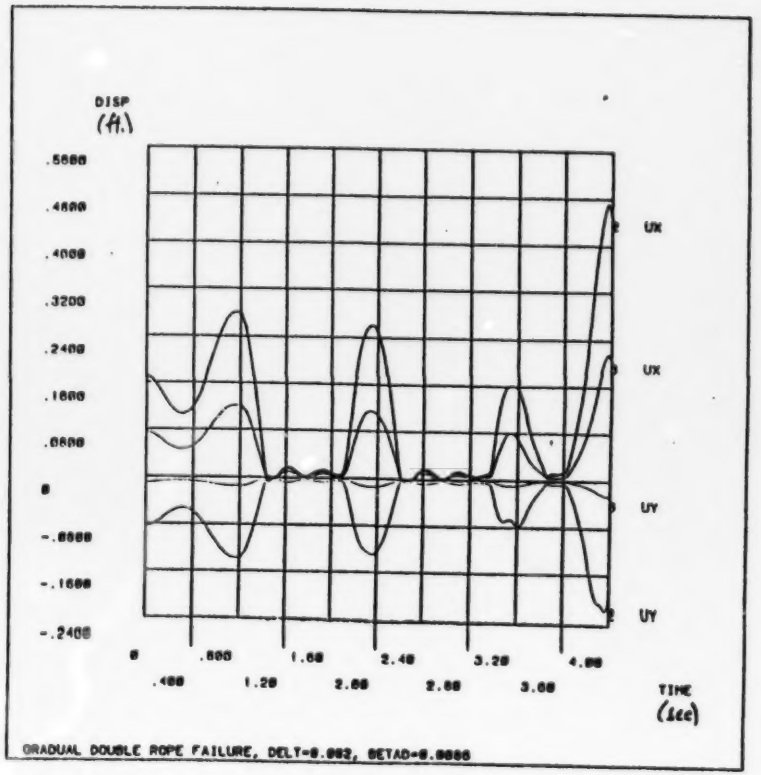


Figure 7(b) - Case 4: Tristructure Nodal Displ.

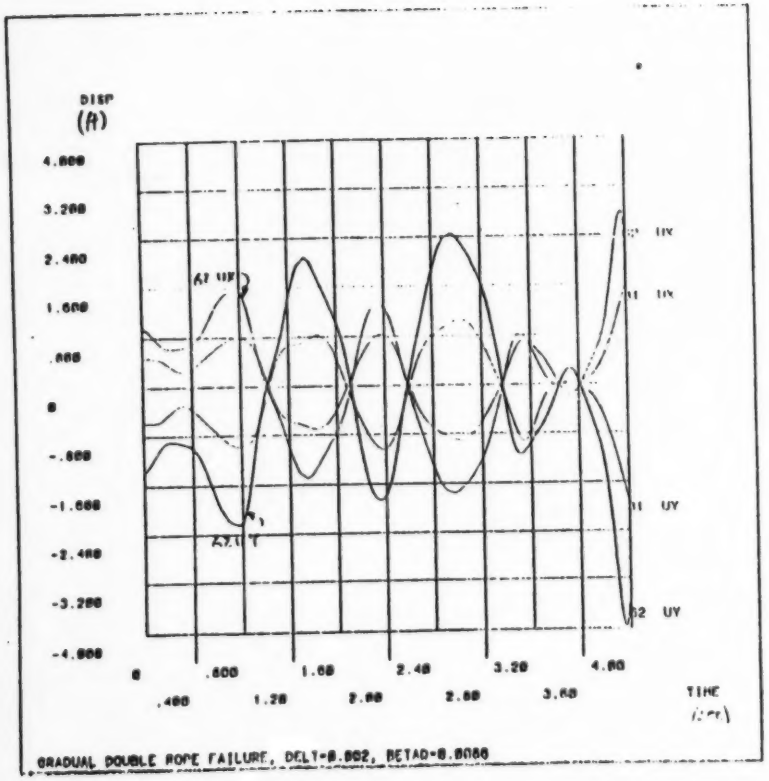


Figure 7(c) - Case 4: Boom Displ.

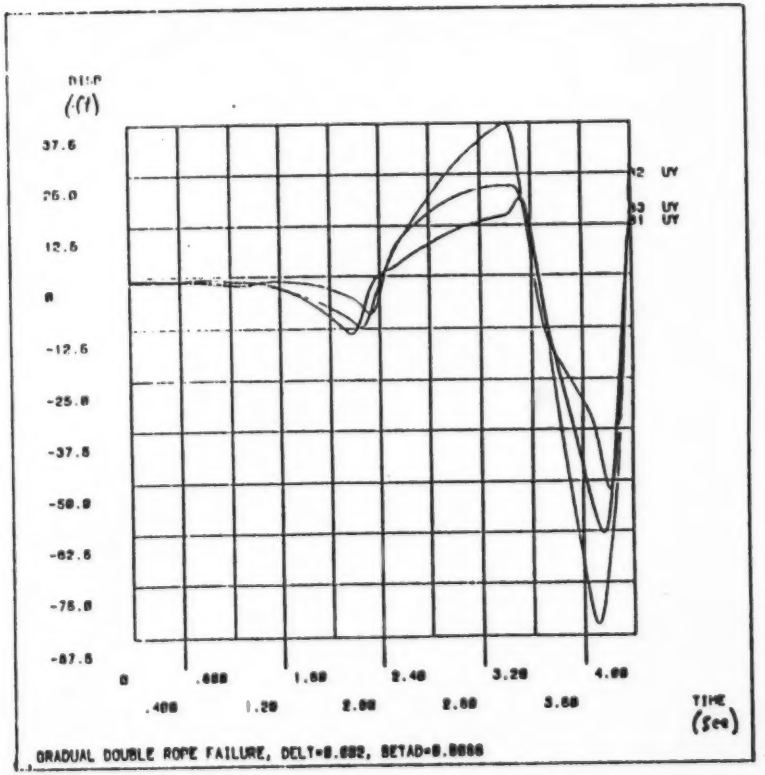


Figure 7(d) - Case 4: Upper Cable Displ.

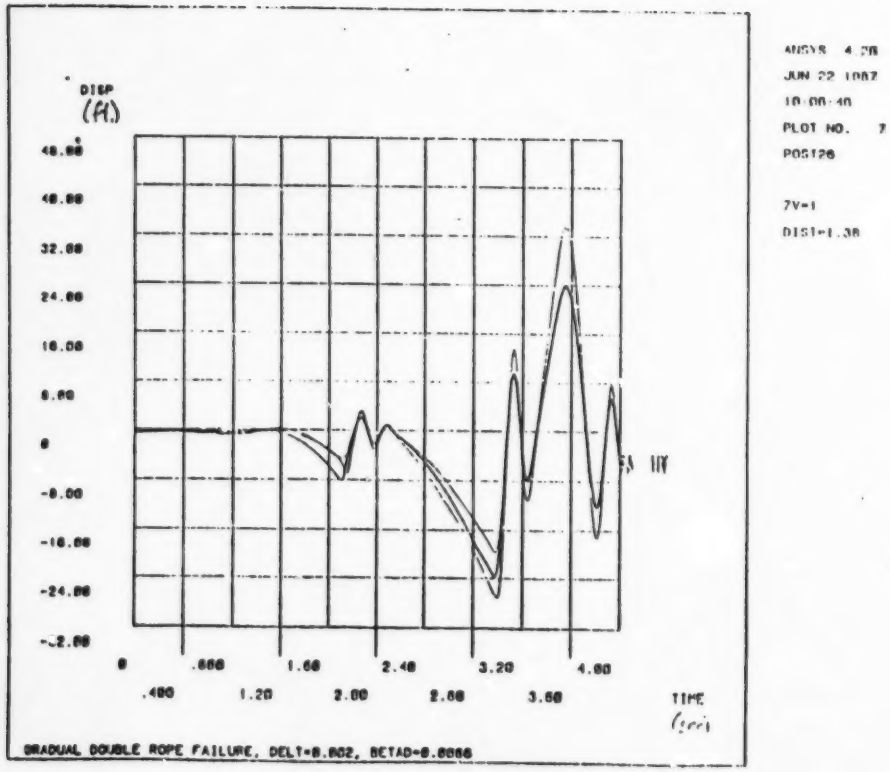


Figure 7(e) - Case 4 - Mid Cable Vert. Displ.

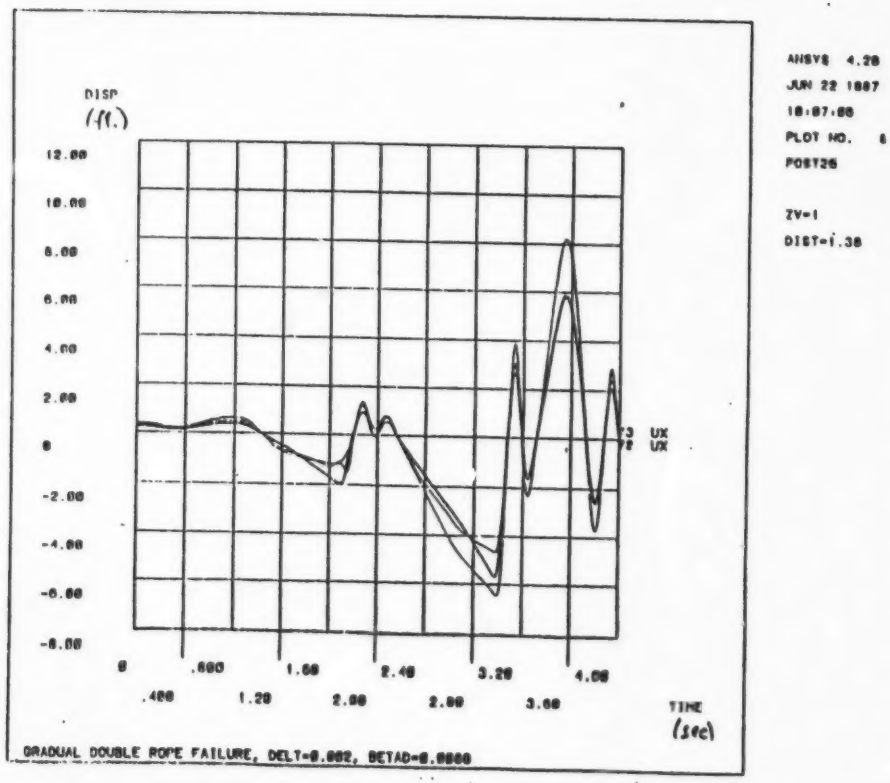


Figure 7(f) - Case 4: Mid Cable Horiz. Displ.

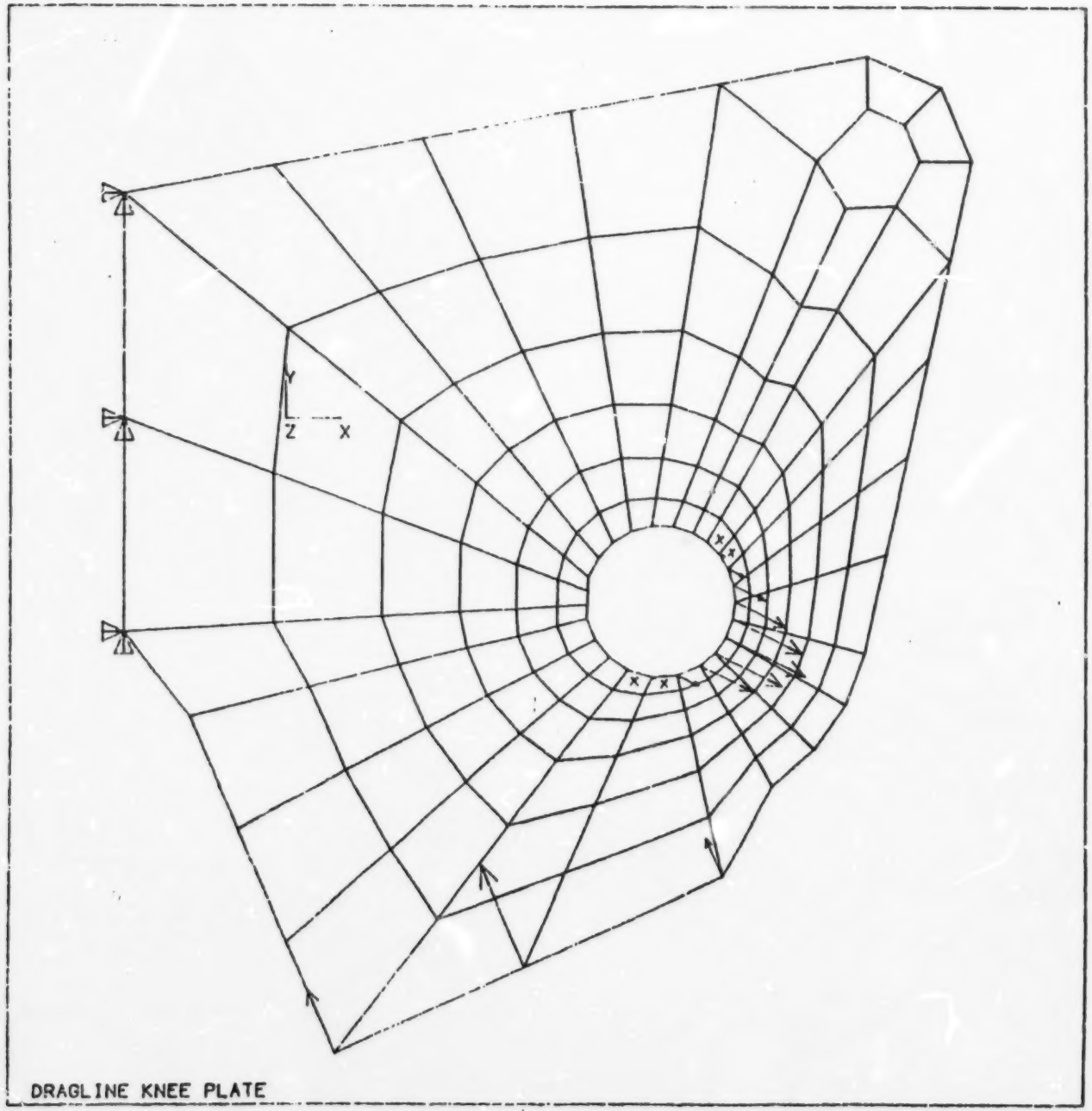


Figure 8 - Finite Element Model of Knee Plate

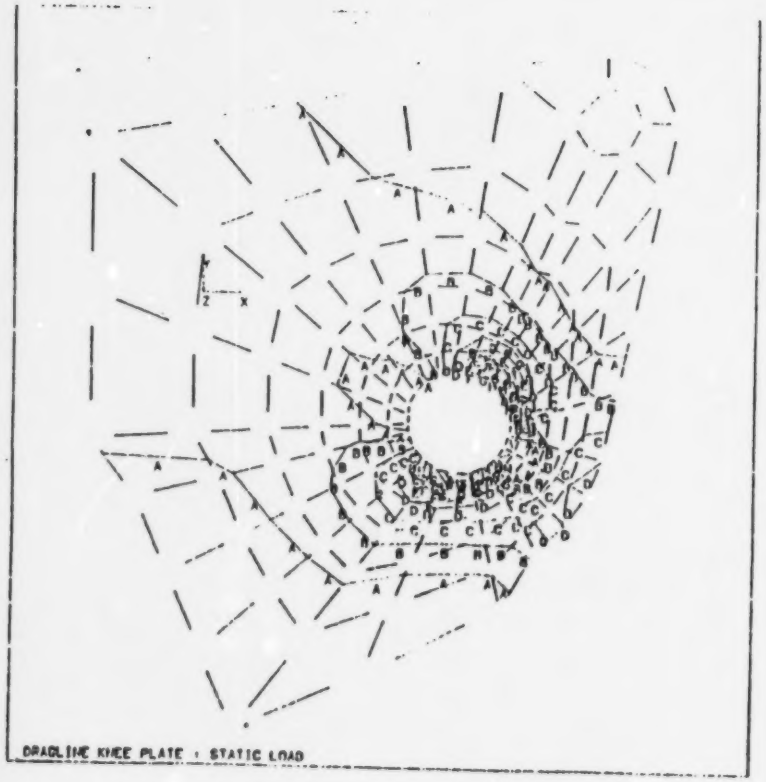


Figure 9(a) - Tensile Stress Contours for Static Loading

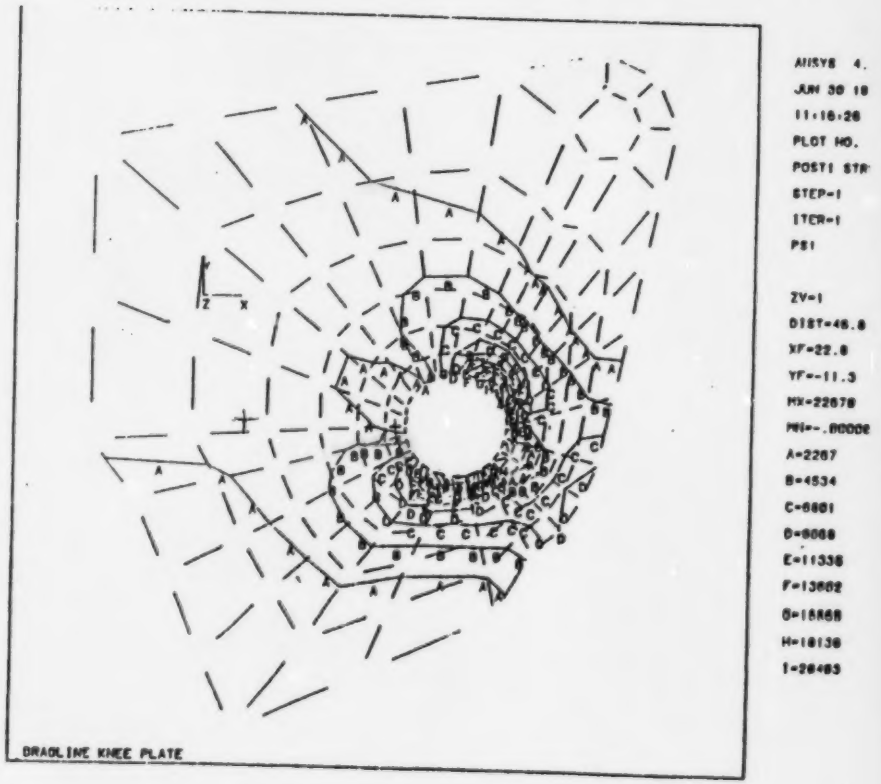


Figure 9(b) - Tensile Stress Contours for Peak Dynamic Loading

Appendix B. User Manual

USER MANUAL:
DRAGLINE MONITOR
(MARION 8200)

Prepared for: Mr. A. Cormier
President and General Manager
N.B. Coal Ltd.
Minto , NB

By: Ray St-Laurent, P.Eng.
Senior Engineer, Electronics
New Brunswick Research & Productivity Council
Fredericton, New Brunswick
Telephone: (506) 452-8994
FAX: (506) 452-1395
Telex: 014-46115

Revision 2
March 30, 1989

OVERVIEW

The NB Coal Dragline Monitor system is designed to monitor the stresses on four selected strain gauged members of the Marion 8200 dragline. Data is recorded on replaceable 3.5 inch diskettes. Two types of records are stored on the diskettes:

1. Routine records are recorded every hour. These contain the minimum and maximum stresses noted over the preceding hourly period.

2. The second record type is an event record. When the stress on any selected member exceeds a specifiable limit, a complete record of the stresses for each member is recorded for each second of the 40 seconds before the overstress, and 20 seconds following.

The micro-computer based monitoring equipment is housed in a wall-mounted unit in the right hand operator's booth of the Dragline. This is connected to strain gauges on particular structural members. An alarm horn and light in the left hand operator's cab is also connected to the monitoring equipment.

A view-port on the monitor displays summary information and plots the stress curves of events.

STARTUP

The system automatically starts itself when it is turned on. The complete instructions are:

1. If not already present, place a writable disk in the slot for the lower disk drive of the computer (it is the only one accessible). This disk will be referred as the LOG disk. The write protect tab must be covering the hole in the corner of the disk.

2. Turn on the computer using its on/off switch. This is a slide switch located at the left side of the upper edge of the computer.

***** IMPORTANT *****

IF THE FOLLOWING MESSAGE IS DISPLAYED, THE PROPER ALARM OVERSTRESS LIMITS HAVE BEEN LOST:

Error reading disk for stress trigger limits.

EXAMINE AND CHANGE THE VALUES AS DESCRIBED IN THE SECTION "Gauge Limits". CALL RPC FOR A PERMANENT REMEDY.

REPLACING DISKS

The display shows the amount of space remaining on the LOG disk. To replace a disk, press the space bar on the computer. Then follow the instructions on the screen. (To test this feature you may reinsert the same disk when it calls for a new disk.)

***** IMPORTANT *****
FAILURE TO FOLLOW THE INSTRUCTIONS ON THE SCREEN CAN CAUSE DISK INFORMATION DAMAGE.

***** IMPORTANT *****
TO MAKE ROOM FOR NEW DATA, AN EXISTING LOG FILE WILL BE ERASED WHEN THE DISK IS INSERTED.

RUNNING DISPLAY

Top Left Shows when the LOG disk file was installed.

Top Right Shows number of overstress events and the space left on LOG disk.

Top Centre Small flashing face indicating the system is active.

Routine record A new line is displayed when each routine hourly report is stored on the LOG disk.

Event Record A new line is displayed whenever an overstress event is stored on the LOG disk.

The word **ALARM** appears at the top centre of the display whenever an overstress occurs. It is removed by pressing the **Last Event** button.

CONTROL BUTTONS

The system comes with three external buttons. Two are on the monitor box. The third is on the alarm horn/light box in the operator's cab.

Alarm Button This is used to test the system. If the system is operating normally, pressing it will activate the horn and alarm light for 2 seconds.

Last Event Plots the stresses recorded in the last overstress event. The curves cover the 60 second period that started 40 seconds before the alarm occurred.

Summary Lists the times of all events stored on the LOG file. If the bottom of the screen is reached before finishing, the following message will be displayed:

More ? (y/n)

Press the **Y** key to see more events.

ABNORMAL MESSAGES

The horn and alarm light will activate when an overstress event occurs. The word **ALARM** will appear at the top centre of the display. The horn will normally cease after about 20 seconds, unless the alarm condition persists. If the event occurs immediately following the installation of a LOG disk, the horn may sound for over 60 seconds.

Two abnormal operating conditions will lead to error messages:

Recalibrate # The strain gauge readings are calibrated to read the correct values. RPC should be called for re-calibration if this message occurs. It is expected that the system can operate for up to a year before re-calibration is necessary.

Powerline Failure The computer has batteries which can automatically power itself for short periods if the AC power is removed. The strain gauge instrumentation is powered directly from AC power. Its interruption causes an appropriate message to be displayed. The computer then stores any pending data to the LOG disk and terminates. This is to prevent the potential corruption of data if the disk could be accessed after batteries were exhausted.

If this message occurs, repair the power failure and restart the program by using the on/off switch on the computer. The program will not restart automatically after a power outage.

***** IMPORTANT *****

TO AVOID DRAINING THE BATTERIES, THE COMPUTER SHOULD BE TURNED OFF IF A POWER OUTAGE LASTS MORE THAN 30 MINUTES. IF THE OUTAGE LASTS MORE THAN 3 DAYS, CALL RPC FOR LONG TERM STORAGE OF THE COMPUTER.

ADVANCED OPERATION

Extra features are available if the monitor program is temporarily halted. Hold the key labelled **"Ctrl"** and press the key **"Z"**. Note that the flashing face stops.

Gauge Limits

There are maximum negative and positive stress limits for each strain gauge. When an observed stress exceeds any limit, an overstress event alarm occurs.

The present limits are displayed by typing **LIMITS** followed by pressing the key **"Enter"**. The computer responds by displaying the negative and positive limits, in pounds per square inch, for each gauge.

The following example shows how to change the limits for gauge number 2:

-10000 17500 2 LIMIT "Enter"

where "Enter" is the key labelled as such. The above sets the alarm levels for gauge number 2 to be below -10000 psi (-68900 kPa) and above 17500 psi (121000 kPa). Note that there must be at least one space between each number or word and no space after the minus sign. A plus sign is optional. Note that words must be in capital letters.

***** IMPORTANT *****
TO PRESERVE ANY CHANGES, THE CURRENT VALUES MUST BE DISPLAYED BY TYPING "LIMITS" FOLLOWED BY PRESSING THE "Enter" KEY.

Other Commands

The following commands are activated by typing the single word, in capitals, followed by pressing **"Enter"**.

MORE Continues the monitor program after an interruption. The face will flash. Use this word after setting the limits.

Even while the monitor program was halted, gauge data was still gathered and stored in a temporary location. If **MORE** is entered within 5 minutes of stopping the monitor program, no data will be lost.

GO Restarts the monitor program. A new datafile will be created on the LOG disk. Any data in the temporary location is lost. The command **MORE** is preferred over **GO** to preserve all data.

-HORN Prevents the alarm horn from sounding.

+HORN Enables the alarm sound. This is the default mode.

-STATUS Does not check the status of ground fault, gain drift, and powerline.

+STATUS Checks for status of ground fault, gain drift, and powerline. Default mode.

ROUTINE Displays the minimum and maximum stresses for each gauge recorded in the hourly reports.

CLEAN Erases any existing files on a LOG disk. This gives the program the complete use of the disk.

This is similar to the "Replace Disk" feature available while the monitor program is running (press the space bar). However, "Replace Disk" only erases a single existing file.

FORMAT This formats LOG disks. Computers require disks to be formatted before they can be used. When disks are purchased, they arrive unformatted. The **FORMAT** command lets you format disks without requiring a separate computer. Formatting automatically leaves the disks in a **CLEAN** state.

Appendix C. Program Listing

```
1200
1201
1202
1203
1204
1205
1206
1207 \ Sealer
1208 \ Turnkey startup screen
1209 \ Dragline monitoring
1210 \ Port access: AIO8
1211 \ Analog port watch
1212 \ Port access: AIO8
1213 \ Queue examination
1214
1215 \ Queue buffer
1216 \ Queue setup and filling
1217 \ Port access: AIO8
1218 \ Analog port watch
1219 \ Task preparation
1220 \ Timer bios interrupt with AIO8 watcher
1221 \ File utilities
1222 \    File utilities
1223 \ Log file utilities
1224 \ Log disk file replacement
1225
1226 \ Regular storage
1227 \    Regular storage
1228 \ Alarm tools
1229 \ Alarm conditions
1230 \ Pending alarms
1231 \ Alarm recognition and recording
1232 \ Alarm setup
1233 \ Presets file
1234 \ Setup and use
1235 \ Monitor
1236 \ I/O line control
1237 \ I/O status lines
1238 \ I/O status lines
1239 \ Plotting dimensions
1240 \ Plotting tools
1241 \ Labelling
1242 \ Plot setup
1243 \ Stress vs time
1244
1245 \ Display aids
1246 \ Banner
1247 \ Event summary
1248 \ Routine records summary
1249
```

1206

1207

```
 0 \ Sealer
 1 HEX   : SEALED   0007 CONTEXT ! ;
 2
 3 0017 VOCABULARY DRAGCOAL    DECIMAL
 4 DRAGCOAL DEFINITIONS
 5 ~ : EXIT   FORTH ;
 6 : GO   GO ;    : MORE   MORE ;
 7 : GAIN   GAIN ;
 8 : LIMIT   LIMIT ;     : LIMITS   LIMITS ;
 9 : -HORN   -HORN ;     : +HORN   +HORN ;
10 : +STATUS   +STATUS ;     : -STATUS   -STATUS ;
11 : ROUTINE   EXTERNAL ROUTINE ;
12 : CLEAN   A: " DEL B:*.*"" COUNT shell DROP  B: ;
13 : FORMAT   A: " FORMAT B:" COUNT shell DROP  B: ;
14 : DIR   DIR ;    : BYE   BYE ;
15
```

1208

```
 0 \ Turnkey startup screen
 1 FORTH DEFINITIONS
 2 : >msecs ( h m s 100ths - d)  >R >R >R
 3    ( hours) 60 U* R> ( mins) M+  60 T* DROP R> ( secs) M+
 4    100 T* DROP R> ( 100ths) M+  10 T* DROP  ( msecs) ;
 5
 6 : PRECOAL   GET-TIME >msecs  T/MS M*/  TICKS 2!
 7    ( A: " DIR B:" COUNT shell DROP )
 8    B: PAGE  SEALED  +STATUS GO ;
 9
10
11
12
13
14
15
```

*** N.B. TURNKEY PROGRAM CREATION NOTES ***

There appears to be a need to access the disk before
an initial file creation. "B: DIR" can do this.
TICKS must be set with the current time, from DOS. Otherwise
the time of creation of the turnkey program will be used.

The procedure followed after loading this application was:

1. UNMAP all units except 0 (FORTH.SRC).
2. TURNKEY LOAD
3. STARTER PRECOAL
4. PROGRAM B:COAL.COM (disk was in drive B:)

1209

```
0 \ Dragline monitoring
1 32 256 48 BACKGROUND REPORTER   REPORTER BUILD   GILD
2 609 LOAD   GRAPHICS LOAD
3 31 WIDTH 1+ C!
4 12 +B  14 +B THRU  10 +B LOAD   ( log file aids, tools)
5 7 +B  9 +B THRU                 ( service a>d)
6 36 +B  38 +B THRU               ( log file status summary)
7 27 +B LOAD                      ( I/O lines control)
8 17 +B  22 +B THRU               ( recordings)
9 30 +B  34 +B THRU               ( Plots)
10 28 +B  29 +B THRU              ( status/auxiliary  lines use)
11 15 +B LOAD                     ( log file setup)
12 23 +B  26 +B THRU              ( Initialise)
13 4 +B LOAD                      ( queue query tools)
14 39 +B LOAD                     ( routine reports)
15 -2 +B  -1 +B THRU              ( turnkey starter)
```

Load screen for dragline monitoring system.

The task REPORTER is built here. It will handle reading strain gage values and placing them in a circular queue.

1210

```
0 \ Port access: AIO8
1 HEX
2  300 CONSTANT PORT
3 : CHOOSE ( n)   [ PORT 2+ ] LITERAL  OUTPUT ;
4 : A>D ( - n)   10  [ PORT ] LITERAL  INPUT  OVER /
5     [ PORT 1+ ] LITERAL  INPUT  ROT *  + ;
6 \ : VALUE ( - n)   0  [ PORT 1+ ] LITERAL  OUTPUT
7 \  BEGIN  [ PORT 2+ ] LITERAL  INPUT  80 AND NOT  UNTIL  A>D ;
8 \ DECIMAL EXIT
9 VARIABLE #PAUSE
10 : READY   #PAUSE @  0 DO LOOP ;
11 : VALUE ( - n)   0  [ PORT 1+ ] LITERAL  OUTPUT READY  A>D ;
12 DECIMAL
13 10 #PAUSE !
14
15
```

1211

```
0 \ Analog port watch
1 2VARIABLE WHEN   VARIABLE PRECISION
2 : INTERVAL ( n)   T/MS */  PRECISION ! ;
3 : !WHEN ( d)   PRECISION @  M+  WHEN 2! ;
4 : NOTE ( d)   2 !QUEUE  0 4  DUP >R  1-
5    DO  I CHOOSE  VALUE  -1 +LOOP  R> !QUEUE  ;
6 : WATCH ( a)   ( ACTIVATE) @TIME  2DUP WHEN 2!  !WHEN  BEGIN
7    @TIME  2DUP WHEN 2@  D< NOT  IF  2DUP NOTE  !WHEN
8    ELSE  2DROP  THEN  PAUSE  ?KEY UNTIL  ( AGAIN) ;
9 1000 INTERVAL
10
11 : TN  VACANT COUNTER 200 0 DO @TIME 2DUP NOTE !WHEN  LOOP
12  TIMER ;
13
14
15
```

INTERVAL sets the port reading interval in milliseconds.

1212

```
0 \ Port access: A108
1 HEX
2 : MES COUNTER  2710 0 DO 0 301 OUTPUT ( 1 0 DO) [ 302 ] LITERAL
3  INPUT          80 AND NOT  IF LEAVE THEN ( LOOP ) LOOP  TIMER
4 ;                                DECIMAL
5 : 6READY 6 0 DO LOOP ; ( ..137 us delay) : 4READY 4 0 DO LOOP ;
6 : 10READY 10 0 DO LOOP ; ( ~190 us del.) : 5READY 5 0 DO LOOP ;
7 : 7READY 7 0 DO LOOP ;   : 8READY  8 0 DO LOOP ;
8 : READY4  1 DROP ;    \ *****  USE FOR 50 MICROSEC DELAY
9 : 10MES COUNTER 10000 0 DO 10READY LOOP TIMER ;
10 : 4MES COUNTER 10000 0 DO 4READY LOOP TIMER ;
11 : 5MES COUNTER 10000 0 DO 5READY LOOP TIMER ;
12 : 6MES COUNTER 10000 0 DO 6READY LOOP TIMER ;
13 : 7MES COUNTER 10000 0 DO 7READY LOOP TIMER ;
14 : 8MES COUNTER 10000 0 DO 8READY LOOP TIMER ;
15 : DMES COUNTER 10000 0 DO READY4 LOOP TIMER ;
```

1213

```
0 \ Queue examination
1 : .Q ( n)   0 DO  QUEUE  I 2* +  ? LOOP ;
2 : .QT ( n)   0 DO  QUEUE  I 12 * +  2@ D.  LOOP ;
3 : .QI ( n i)   2* 4 + SWAP   0 DO  QUEUE  I 12 * +  OVER + ?
4    LOOP DROP ;
5 : ?Q   'QUEUE @  QUEUE - . ;
6
7
8
9
10
11
12
13
14
15
```

.Q prints 'n' queue cells.

.QT prints 'n' time values (assumes event entries).

.QI prints 'n' values for gage 'i' (assumes event entries).

?Q shows the next queue location to be written.

1214

1215

```
0 \ Queue buffer
1 CREATE QUEUE  HERE 5000  DUP ALLOT  ERASE
2 VARIABLE 'QUEUE
3 : VACANT   QUEUE 'QUEUE ! ;
4 : !QUEUE ( ?v n)   0 DO   'QUEUE @ !  2 'QUEUE +!  LOOP ;
5
6
7
8
9
10
11
12
13
14
15
```

1216

```
0 \ Queue setup and filling
1 VARIABLE 'QUEUE    VARIABLE QHEAD
2 4800  CONSTANT QLEN
3 CREATE QUEUE  HERE QLEN  DUP ALLOT  ERASE
4 : VACANT   QUEUE  DUP 'QUEUE !  QHEAD !  QUEUE QLEN ERASE ;
5 : !QUEUE ( n)   [ ASSEMBLER ]   0 POP   IS SEG  'QUEUE W MOV
6    STOS   W 'QUEUE MOV   QUEUE QLEN + #  'QUEUE CMP
7    0= IF   IS SEG  QUEUE #  'QUEUE MOV   THEN  ;
8 VACANT
9
10 CODE QTEST !QUEUE NEXT
11
12
13
14
15
```

'QUEUE holds the address of the next location in the circular buffer (logical tail of queue).

QHEAD holds the logical head of the circular queue. It should be updated by words elsewhere that process queue entries.

!QUEUE stores a number in the queue and advances the pointer. It handles wraparound when the physical end of the queue is reached.

VACANT erases the buffer and resets QHEAD and 'QUEUE to match the physical beginning at QUEUE.

NB. This is a low level implementation.
Colon definitions are assembler macros.

1217

```
0 \ Port access: AIO8
1 HEX   300 CONSTANT PORT    CVARIABLE OUTMASK
2 VARIABLE #PAUSE   28 #PAUSE !
3 : INP ( - n)   [ ASSEMBLER ] 2 MOV   0 0 SUB   (2) IN  0 PUSH ;
4 : OUTP ( n)   [ ASSEMBLER ] 2 MOV   (2) OUT ;
5 : A>D ( - n)   [ ASSEMBLER ]   PORT # INP   W POP   4 # 1 MOV
6    W SHR V   PORT 1+ # INP   0 POP   4 # 1 MOV
7    0 SHL V   W 0  ADD   0 PUSH ;
8 : slow   [ ASSEMBLER ]  0 PUSH   0 POP ;
9 CREATE ready   IS SEG  #PAUSE 1 MOV   BEGIN  slow  LOOP   RET
10 : VALUE ( - n)   [ ASSEMBLER ]  PORT 1+ # OUTP
11    ready CALL   A>D ;
12 : CHOOSE ( n)   [ ASSEMBLER ]  OUTMASK LDA B   F8 # 0 AND
13    W POP   W 0 OR   OUTMASK STA B    PORT 2+ # OUTP ;
14 CODE READING ( - n)   VALUE NEXT
15 CODE CHANNEL ( n)   CHOOSE NEXT     DECIMAL
```

PORT is the installed I/O address of the AIO8 board.
OUTMASK holds the mask for output.
CHOOSE selects the a>d channel

VALUE initiates and reads the chosen channel.

ready is used to ensure the conversion process is finished before the channel is read.

OUTP expects the value to be output in register 0.

NB. This is a low level implementation.
Colon definitions are assembler macros.

READING is a CODE word that returns VALUE.
CHANNEL is a CODE word that sets the channel with CHOOSE.

1218

```
0 \ Analog port watch
1 VARIABLE keep       VARIABLE PRECISION
2 ~ VARIABLE DATA     VARIABLE NOTED
3 : INTERVAL ( n)   T/MS */  PRECISION ! ;
4 ASSEMBLER
5 CODE !watch   IS SEG  TICKS #  W MOV   2 W) PUSH  W ) PUSH
6    ( STI)   !QUEUE  !QUEUE   4 # 1 MOV   BEGIN   IS SEG
7    1 keep MOV   1 0 MOV   0 DEC   0 PUSH   CHOOSE  slow  VALUE
8    keep 1 MOV   LOOP   4 # 1 MOV   BEGIN  !QUEUE  LOOP   NEXT
9 : WATCH   @TIME DROP  PRECISION @  /  DUP NOTED @ -
10   IF   NOTED !  !watch  1 DATA +!   ELSE   DROP  THEN ;
11
12 : GAZE ( a)   ACTIVATE  BEGIN  WATCH  PAUSE AGAIN ; RECOVER
13 FORTH   1000 INTERVAL
14
15
```

!watch stores 12 bytes in the circular queue.
The current TICKS are stored as a double value.
The 4 a>d chanels are stored as single numbers.
WATCH calls !watch when time changes by PRECISION amount.
INTERVAL sets the a>d reading interval in milliseconds.
It sets the value in PRECISION.
DATA is incremented when a sample is placed on the queue.
NOTED holds the low word of the latest time serviced.

GAZE activates a task to perform a continuous WATCH.

1219

```
0 \ Task preparation
1 VARIABLE 'REPORT
2 : PAUSE?   0. 1 handle @ -seek DROP   'REPORT @EXECUTE
3              0 handle @ -SEEK DROP  ;
4 : Y/N ( - t)   ." ?  (y/n) "  BEGIN  PAUSE? ?KEY UNTIL  (key)
5    DUP EMIT SPACE  32 OR [ASCII] y = ;
6 2047 CONSTANT ZERO
7
8
9
10
11
12
13
14
15
```

'REPORT holds address of report generator.
PAUSE? pauses and allows the report generator to execute once.
The original file position is restored before exiting.
Y/N executes PAUSE? while waiting for keyboard input.
ZERO returns the value corresponding to an analog zero.

1220

```
0 \ Timer bios interrupt with AIO8 watcher
1 ASSEMBLER
2 BEGIN   DS PUSHS  ES PUSHS  0 PUSH 1 PUSH
3    2 PUSH  I PUSH  W PUSH  U PUSH   0 IS SSG  0 DS LSG  STD B
4    IS SEG  TICKS 2+ INC   0= IF
5    IS SEG  TICKS INC   THEN   WATCH
6    U POP  W POP  I POP  2 POP 1 POP  0 POP  ES POPS  DS POPS
7 28 INTERRUPT    DECIMAL
8
9
10
11
12
13
14
15
```

This code replaces the BIOS timer tick interrupt.
It modifies the standard code by appending WATCH.

For success it is important to:
Save the registers and segments shown here.
Set the DS to match IS segment.
Invoke STD B to set the direction flag.

NB. This is a low level implementation.
Colon definitions are assembler macros.

NB. The register U must be pushed last on the stack for WATCH

*** NOT USED DUE TO INTERRUPT TIMING CONTENTIONS ***
(NB.- WATCH is also no longer compatible; U not used)

1221

```
0 \ File utilities
1 HEX
2 : -FCREATE ( a - fh f)  (D !DATA  0 !EXT  3C -DOS
3    RCODE D)  SWAP ;
4 : -DELETE ( a - f)   (D !DATA  41 -DOS D) ;
5 : -FOPEN ( a b - fh f)  (D !FC  !DATA  3D -DOS
6    RCODE D) SWAP ;
7 : -FCLOSE ( fh - f)  DUP
8    0< IF  0= ELSE  (D !FH  3E -DOS D)  THEN ;
9 : -SEEK ( d n fh - f)   (D !FH  !FC  !EXT  !DATA  42 -DOS D) ;
10 : -COMMIT ( fh - f)   (D !FH  68 -DOS D) ;
11 : -FILE ( az - f)   PAD match ;
12 : -seek ( d n fh - d f)   -SEEK  >R RCODE PARAMS 4 + @  R> ;
13 : >END ( fh)   FSIZE? 2DROP ;
14 DECIMAL
15
```

These words correspond to DOS functions with similar names.
The -XXXXX nomenclature means a true flag is returned if
the operation fails.

Input addresses begin null terminated ASCII strings,
such as via NAME or >ASCIZ.

-SEEK is a seeks a file position from the beginning.
-seek returns the actual file pointer position as well.
-FILE looks for a file.
>END moves a file pointer to the end of a file.
-COMMIT writes buffers to disk.

1222

```
0 \    File utilities
1 HEX
2 VARIABLE handle
3 : -gets ( a n fh - n f)   (D !FH  !EXT !DATA  3F -DOS
4    RCODE D)  SWAP ;
5 : -puts ( a n fh - n f)   (D !FH  !EXT !DATA  40 -DOS
6    RCODE D)  SWAP ;
7 : -PUTS ( a n fh - f)   OVER >R  -puts  SWAP R> -  OR ;
8 : -GETS ( a n fh - f)   OVER >R  -gets  SWAP R> -  OR ;
9 : PUTS ( a n fh - n)   (D !FH  !EXT !DATA  40 DOS RCODE D) ;
10 : DRIVE? ( n - b/s s/c c c/d)   (D !DATA  36 DOS
11    PARAMS >R  R@ 2+ @  RCODE  R@ 6 + @  R> 4 + @ D) ;
12 : !+ ( a n - a')   OVER !  2+ ;
13 : 2!+ ( a d - a')   SWAP >R  !+  R>  !+ ;
14 : .ERR   DISK 4 + ? ;
15 DECIMAL
```

handle stores a file handle.
-PUTS writes 'n' bytes to disk starting from 'a'.
-GETS reads 'n' bytes from disk to 'a'.
-puts and -gets also return actual bytes transferred.
PUTS relies on DISK 2+ to record accumulated disk use errors.
.ERR prints the current DOS disk error code.
DRIVE? has drive number for input...0=default; 1=A, etc.
returns bytes/sector sectors/cluster available clusters
and clusters on disk.

!+ store a value at an address then increment the address.
2!+ as above for double numbers.

1223

```
0 \ Log file utilities
1 VARIABLE precision                ( milliseconds)
2 VARIABLE PERIOD      60 PERIOD !   ( minutes)
3 12 CONSTANT SAMPLE    60 CONSTANT #SAMPLES
4 SAMPLE #SAMPLES *  CONSTANT ENTRIES
5 VARIABLE #EVENTS
6 CREATE PORTAL   24 ALLOT
7 : !DAYTIME ( a - a')   @DATE  !+  @TIME 2!+ ;
8 : name# ( - d)   GET-DATE  100 MOD  100 * +  100 M*  ROT M+
9    100 1 M*/  GET-TIME DROP 2DROP  M+ ;
10 : name ( - az)  name#  <# # # # # # # # # #> >ASCIZ
11    " .LOG" COUNT  PAD 8 + SWAP  CMOVE   0 PAD 13 + C! ;
12 : PREPARE ( - f)   " *.LOG" COUNT >ASCIZ
13    PAD match DROP  PAD 30 +  -DELETE ;
14
15
```

precision holds sampling interval in milliseconds.
PERIOD holds the regular recording interval in minutes.
#SAMPLES number of samples saved for an event.
ENTRIES number of bytes required for an event.
#EVENTS number of events recorded on the current disk.
!DAYTIME stores current date and time, then increments queue.
name returns an ASCIZ file name with date and hour encoded
YYMMDDHH.LOG
PREPARE removes an existing LOG file.

1224

```
 0 \ Log disk file replacement
 1 : OHEADER ( - f)   PORTAL  DUP 24 ERASE  precision @ !+
 2    PERIOD @ !+   !DAYTIME DROP  PORTAL 24  handle @  -PUTS ;
 3 : NEWLOG ( - f)   name -FCREATE  DUP  IF  SWAP DROP
 4   ELSE  DROP   handle !  PRECISION @  T/MS SWAP */  precision !
 5   OHEADER   handle @  -COMMIT  OR   THEN ;
 6 : ENDLOG ( - f)   10. 0 handle @  DUP >R -SEEK  PORTAL !DAYTIME
 7    #EVENTS @ SWAP !  PORTAL 10  R@ -PUTS  R> -FCLOSE  OR OR ;
 8 : ?tab   ."   Check write protect tab." ;
 9 : .ERRSET   .ERR ." Error setting up file." ?tab ;
10 : REPLACE   CR ." Working..."  ENDLOG
11    IF  CR .ERR ." Error closing file."  THEN  BEGIN
12    CR ."   Replace disk.  Press 'Enter' key when finished"
13    (KEY) DROP  PREPARE DROP  NEWLOG DUP
14    IF  CR .ERRSET   THEN  0= UNTIL   BORN  PAGE BANNER ;
15
```

ENDLOG updates the LOG file header before closing the file.
NEWLOG creates a new LOG file and store the initial header
OHEADER writes the initial header to the log file.

REPLACE is used to change the LOG disk.
The user is prompted.

1225

1226

```
 0 \ Regular storage
 1 CREATE MIN/MAX   16 ALLOT
 2 4
 3 : date   name# 1 100 M*/  <# # #  [ASCII] / HOLD  # #
 4    [ASCII] / HOLD  # # #>  TYPE SPACE ;
 5 : OF ( a n - a)   2* + ;
 6 : RESET   [ DUP ] LITERAL  0 DO  I CHANNEL READING
 7    MIN/MAX  I 2* OF  2DUP 2+ ! !  LOOP ;
 8 : !MIN/MAX   MIN/MAX  'QUEUE @ [ DUP 2* ] LITERAL -
 9    DUP QUEUE < IF  QLEN + THEN   LITERAL 0 DO
10      2DUP I OF @  SWAP  I 2* OF  SWAP >R   DUP 2+ DUP @
11      R@ MAX  SWAP !   DUP @  R> MIN  SWAP !   LOOP 2DROP ;
12
13
14
15
```

MIN/MAX holds the minimum and maximum recording interval
values for each channel.
!MIN/MAX   updates MIN/MAX for the latest queue sample.

RESET allows for a zero value corresponding to ZERO offset.

date prints current date in format YY/MM/DD.
OF allows array type access knowing the starting address.

1227

```
0 \    Regular storage
1 2VARIABLE RECORDED   -1. RECORDED 2!
2 : ?HOUR ( - f)   @TIME T/MIN U/MOD  PERIOD @ /
3    RECORDED 2DUP  @ -  DUP >R  IF  >R  24 MOD  R@ !
4    @DATE R> 2+ !   ELSE  2DROP THEN   DROP R> ;
5 : REGULAR   monochrome  PORTAL  -1 !+  !DAYTIME  MIN/MAX SWAP
6   16 CMOVE  handle @ >R  R@ >END  PORTAL 24  R@ -PUTS  R>
7   CR 10 SPACES  -COMMIT  CR CR IF .ERR ." Error "  THEN
8   ." Writing regular report at " date TIME  RESET  .USAGE ;
9
10
11
12
13
14
15
```

REGULAR stores one regular record
?HOUR determines if a recording period has passed
       since the last regular recording.
RECORDED   stores the latest recorded hour.
RECORDED 2+ stores the latest recorded day.

1228

```
0 \ Alarm tools
1 VARIABLE DOOMSDAY     2VARIABLE DOOMSTIME   -1 DOOMSTIME !
2 VARIABLE 'REPLAYS     VARIABLE QALARM
3 VARIABLE 'alarm
4 CREATE REPLAY  ENTRIES ALLOT
5 CREATE DAMN     HERE 8  DUP ALLOT  ERASE
6 CREATE GRACE    HERE 8  DUP ALLOT  ERASE
7 CREATE TRIGGER  HERE 16 DUP ALLOT  ERASE
8 : ?ALARM ( a)   'alarm @EXECUTE ;
9 : CLOCKED ( - d)   QHEAD @ 2@ ;
10 : LOWER ( a i - a)   4 * + ;
11 : UPPER ( a i - a)   4 * + 2+ ;
12 : OVERSTRESS ( a - a)   4 + ;
13 : OVERRUN?   DATA @  [ QLEN SAMPLE /  42 - ] LITERAL  >
14    IF  0 DATA !  'QUEUE @ QHEAD !  -1 DOOMSTIME !  THEN ;
15
```

DAMN holds consecutive violation count (time) for each channel.
GRACE holds allowable count before an alarm called.
TRIGGER holds minimum and maximum permissible values.
DOOMSTIME holds the pending alarm time.
DOOMSDAY holds the pending alarm day.
'REPLAYS holds the address of the current pointer in the
          REPLAY area.
QALARM holds the address of the start of the latest alarm data.
CLOCKED returns the time of the queue area being serviced.
OVERSTRESS converts a sample to the start of the gages.
OVERRUN? resets the queue head and clears a pending alarm
          if unserviced queue is about be overrun by new data.
LOWER accesses minimum trigger word.
UPPER accesses maximum trigger word.
?ALARM holds the alarm recognition code.

1229

```
0 \ Alarm conditions
1 : ?DAMNED ( n i - f)   >R  DUP  TRIGGER R@ LOWER @  <
2    SWAP  TRIGGER R@ UPPER @  >  OR
3    IF  DAMN R@ OF  1 OVER +!  @ GRACE R> OF @  >
4    ELSE  0 DAMN R> OF !  0   THEN ;
5 : ALERT ( a - f)   0 SWAP  4 0 DO  DUP I OF @  I ?DAMNED
6    IF  >R 1+ R>  THEN  LOOP  DROP ;
7 : ALARM ( -f)   CLOCKED  DOOMSTIME 2@  D< NOT
8    RECORDED 2+ @  DOOMSDAY @  < NOT  AND ;
9 : !QALARM   QHEAD @  SAMPLE +  ENTRIES -  DUP QUEUE <
10    IF  QLEN + THEN   QALARM !  REPLAY 'REPLAYS ! ;
11 : REPLAY! ( a n)   >R  'REPLAYS @  R@ CMOVE  R> 'REPLAYS +! ;
12
13
14
15
```

The end time of an event is found in DOOMSDAY and DOOMSTIME.
A negative value for DOOMSTIME indicates no alarm is pending.
REPLAY holds a copy of the current event data.
QALARM is set to the beginning of an event by !QALARM.
?DAMNED determines if a channel's values are exceeded.
ALERT checks each channel for ?DAMNED.
ALARM determines if an event should be recorded now.
REPLAY! copies event data into the REPLAY display area
         and updates the 'REPLAYS pointer.

1230

```
 0 \ Pending alarms
 1 2VARIABLE LASTDOOMED
 2 : REBORN   DOOMSTIME 2@  LASTDOOMED 2!
 3    -1 DOOMSTIME !  @DATE DOOMSDAY ! ;
 4 : BORN   0 DATA !  -1. RECORDED 2!  REBORN  0 #EVENTS ! ;
 5 : ?REDEEMED   DAMN DUP 2@ OR  SWAP 4 + 2@  OR OR
 6    0= IF  -1 DOOMSTIME !  THEN ;
 7 : !PENDING   CLOCKED  [ 20000 T/MS */ ] LITERAL  M+
 8    LASTDOOMED 2@  [ #SAMPLES T/MS 1000 / */ ] LITERAL M+
 9    DMAX   2DUP T/DAY  D< NOT  IF  T/DAY D- 1
10    ELSE 0 THEN  @DATE +  DOOMSDAY !  DOOMSTIME 2! ;
11 : -DOOMED ( - f)   OVERRUN?  DOOMSTIME @  0< ;
12 : PENDING   -DOOMED IF  !PENDING   NOISE  THEN ;
13 : ?alarm ( a -f)   OVERSTRESS ALERT  -DOOMED NOT  OR
14    IF  PENDING ALARM   ELSE  ?REDEEMED 0  THEN ;
15
```

LASTDOOMED stores the previous alarm time.
REBORN resets alarm time values.
BORN is used for a new file to reset alarm time values, resets the regular recorded time, and zeroes the queue DATA requried to be serviced.

-DOOMED returns a true flag if no alarm is pending.
?REDEEMED resets DOOMTIME if no DAMN values are set.
PENDING sets the time for the end of an event
?alarm determines if that address is in violation.
It is normally vectored into 'alarm.

1231

```
 0 \ Alarm recognition and recording
 1 : +SAMPLE ( a)   >R  SAMPLE R@ +!  R@ @  [ QUEUE QLEN + 1- ]
 2    LITERAL >  IF  QUEUE R> !  ELSE  R> DROP   THEN ;
 3 : event ( a fh - f)   >R  REPLAY ENTRIES ERASE
 4    #SAMPLES 0 DO  DUP @  SAMPLE REPLAY!  DUP +SAMPLE  LOOP
 5    DROP  REPLAY ENTRIES R@ -PUTS   R> -COMMIT  OR ;
 6 : REMEMBER   monochrome   handle @  DUP >END
 7    QALARM SWAP event  CR IF .ERR ." ERROR "  THEN
 8    ." RECORDING EVENT AT "  date TIME ;
 9 : EVENT?   BEGIN   PAUSE   OVERRUN?  DATA @   WHILE  -1 DATA +!
10    QHEAD @  ?ALARM   IF  !QALARM   REMEMBER  .ALARM  QUIET
11    1 #EVENTS +!  .USAGE  REBORN   THEN
12    QHEAD +SAMPLE   .ALIVE REPEAT ;
13 : TREMEM   QHEAD  +SAMPLE !QALARM  REMEMBER  ;
14 : TREMS ( n)   0 DO  TREMEM LOOP ;
15
```

REMEMBER saves the event data to disk. It also copies the event data into a secure space.

+SAMPLE updates a queue pointer.

EVENT? handles the identification of events. If so...
event data is written to disk, the replay area is setup
Alarm annunciation is handled.
The QALARM and QHEAD pointers are updated.

The procedure is repeated until serviced data, indicated by QHEAD and DATA, reaches all data placed in the queue, indicated by 'QUEUE.

1232

```
 0 \ Alarm setup
 1 VARIABLE SOFTSTART
 2 : CLEAR   0 OUTMASK C!  0 PORT 2+ OUTPUT
 3    50 SOFTSTART !   'alarm ASSIGN  ( a)   DROP  -1 SOFTSTART +!
 4       SOFTSTART @ 0=  IF  ['] ?alarm  'alarm !  THEN  0 ;
 5
 6
 7
 8
 9
10
11
12
13
14
15
```

SOFTSTART allows alarm conditions on startup to be ignored. This allows the a/d system to stabilize after powerup and sufficient history to be available for recording. 50 seconds are allocated here.

CLEAR clears any alarms and vectors softstarting code into 'alarm. After the sofstart period, the regular ?alarm is vectored into 'alarm.

1233

```
0 \ Presets file
1 CREATE GAIN   8 ALLOT
2 : .PRESETS-ERR   .READ-ERR ."  for stress trigger limits" ;
3 : -presets ( - fh f)   " C:\COAL.SET"  COUNT
4   >ASCIZ  R/W -FOPEN ;
5 : PRESETS   -presets IF  DROP CR .PRESETS-ERR
6   ELSE  >R  PORTAL 24 R@ -GETS   IF CR .PRESETS-ERR
7      ELSE  PORTAL TRIGGER 16 CMOVE   PORTAL 16 +  GAIN 8 CMOVE
8      THEN  R> -FCLOSE DROP  THEN ;
9 : >PRESETS   TRIGGER PORTAL 16 CMOVE   GAIN PORTAL 16 + 8 CMOVE
10   -presets DUP  NOT
11   IF  DROP >R   PORTAL 24 R@ -PUTS  R> -FCLOSE  CR  THEN
12   IF  CR .ERR ." Error saving stress trigger limits" THEN ;
13
14
15
```

The trigger limits and gage gain values are stored in a file.
This allows them to be changed without requiring a recompile of the program.

GAIN holds the gain settings for each gage (0 to 3).

preset returns the file handle, and flag, for the file.
PRESETS reads the values in the file into the running system.
>PRESETS archives current values into the file.

1234

```
0 \ Setup and use
1 : S>D ( n - d)   DUP 0< ;
2 : PSI/GAGE ( i - n1 n2)   GAIN SWAP OF @
3   [ 30 2047 ] 2LITERAL  >R * R> ;
4 : STRESS ( n i+1 - n)   >R  ZERO -  R> 1- PSI/GAGE */ ;
5 : STRESS> ( n i+1 - n)   1- PSI/GAGE SWAP */  ZERO + ;
6 : LIMIT ( lo hi i+1)   DUP >R  STRESS>  SWAP R@ STRESS>
7   TRIGGER R> 1- LOWER 2! ;
8 : LIMITS   5 1 DO  CR TRIGGER I 1- LOWER 2@  SWAP I STRESS SWAP
9   I STRESS  S>D 8 D.R  S>D 8 D.R  ."   Gage # " I .
10   LOOP  >PRESETS ;
11 1000 GAIN 0 OF !       1000 GAIN 1 OF !
12 1000 GAIN 2 OF !       1000 GAIN 3 OF !
13 -30000 30000 1 LIMIT    -30000 30000 2 LIMIT
14 -30000 30000 3 LIMIT    -30000 30000 4 LIMIT
15
```

This screen provides setup values for the strain gages and other parameters.

GAIN holds the gain settings for each gage (0 to 3).

STRESS> converts from psi to internal units.
STRESS provides the converse function.
LIMIT sets TRIGGER values. Minimum, maximum and gage number are input for each gage.
LIMITS displays the current limit values.

NB.- The gages are number 1 to 4 for LIMIT(S), STRESS & STRESS>.

1235

```
0 \ Monitor
1 : MONITOR    EVENT?  !MIN/MAX  ?HOUR  IF  REGULAR  THEN ;
2 ' MONITOR  'REPORT !
3 : NEWDISK ( c - f)   32 = ( Space bar)  IF  REPLACE THEN ;
4 : ABANDON ( c - f)   26 = ( Ctrl-Z) ;
5 : REPORT   BEGIN   PAUSE  MONITOR   STATUS?
6   ?KEY IF  (key)  DUP NEWDISK   ABANDON
7   ELSE   0  THEN  UNTIL ;
8 : go   B: NEWLOG  IF CR .ERR  1 ABORT" Error creating file "
9   THEN   BORN RESET VACANT  REPORTER GAZE
10   PAGE BANNER  CLEAR REPORT ;
11 : GO   CLEAR  PRESETS go ;
12 : MORE   CLEAR REPORT ;
13 +STATUS
14
15
```

MONITOR activates the task to check conditions and then writes a regular record or an alarm event when appropriate.
REPORT repeatedly calls MONITOR, checks the status lines and checks for keyboard input.
A ^Z breaks out of the loop.
Touching the space bar initiates the REPLACE disk word.

GO is the main word that runs the monitoring application.
It creates a file, starts the REPORTER task, sets up the screen and then calls REPORT.

MORE is used to continue monitoring after it has been stopped.

1236

```
0 \ I/O line control
1 16 CONSTANT HORN   32 CONSTANT LIGHT
2 : on ( n)   OUTMASK C@  OR  DUP OUTMASK C!  PORT 2+ OUTPUT ;
3 : off ( n)   -1 XOR  OUTMASK C@  AND
4    DUP OUTMASK C!  PORT 2+ OUTPUT ;
5 : NOISE   HORN on  LIGHT on ;
6 : QUIET   HORN off ;
7 : -HORN   128  ' HORN ! ;
8 : +HORN   16  ' HORN ! ;
9
10
11
12
13
14
15
```

The output words here control output by manipulating OUTMASK.

on resets the bits set in the number on the stack for output.
off sets the bits set in the number on the stack for output.
CLEAR resets all output bits and changes OUTMASK.
?ON checks the status of the bits in the input number against the input port.

HORN leaves the bit pattern for the horn.
LIGHT leaves the bit pattern for the light.

NOISE turns on horn and light.
quiet turns off horn (light remains).
-HORN is used to disable horn.
+HORN enables horn.

1237

```
0 \ I/O status lines
1 VARIABLE 'STATUS
2 : STATUS?   'STATUS @EXECUTE ;
3 : -STATUS   0 'STATUS ! ;
4 : TEST? ( b)   16 AND  0= IF  HORN on  LIGHT on  2000 MS
5    LIGHT -DOOMED AND  HORN OR  off  THEN ;
6 : DISPLAY? ( b)   64 AND  0= IF  LIGHT off  DISPLAY  THEN ;
7 : SUMMARY? ( b)   32 AND  0= IF  EVENTS  THEN ;
8 : ?BUTTONS ( - b)   PORT 2+ INPUT  112 AND ;
9
10
11
12
13
14
15
```

'STATUS holds the addres of code used to check status lines.
-STATUS ignore status lines.

?BUTTONS checks if any buttons pressed, leaves button code.
TEST? if test button pressed, light and horn on for 2 seconds.
DISPLAY? if display button pressed, plot last event.
summary? if summary button pressed, list events on disk.

1238

```
0 \ I/O status lines
1 : CENTIVOLTS ( n - n)   ZERO -  [ 500 2047 ] 2LITERAL  */ ;
2 : ?GROUND ( - f)   5 CHANNEL READING CENTIVOLTS   -25 25 WITHIN
3    NOT DUP IF  CR 15 SPACES  " GROUND FAULT" COUNT MARK  THEN ;
4 : ?GAIN ( - f)  4 CHANNEL READING CENTIVOLTS   451 499 WITHIN
5    NOT DUP IF  CR 15 SPACES  " GAIN DRIFT"  COUNT MARK  THEN ;
6 : -POWER ( - f)   7 CHANNEL READING CENTIVOLTS   250 <
7    DUP IF  CR 15 SPACES  " *** POWERLINE FAILURE ***"
8    COUNT MARK  THEN ;
9 : +STATUS   'STATUS ASSIGN   ?BUTTONS  ?DUP IF  DUP TEST?
10   DUP DISPLAY?   SUMMARY?  ?GROUND DROP  ?GAIN DROP
11   -POWER ABORT"  Program terminated"  THEN ;
12 AKA TEST?  T     AKA DISPLAY?  D   AKA SUMMARY?  S
13 AKA ?GROUND  G   AKA ?GAIN  K      AKA -POWER  P
14
15
```

CENTIVOLTS converts a/d line into centivolts.
?GROUND prints message if ground line fault.
?GAIN prints message if gain reference line out of bounds.
-POWER prints message if powerline failure.

+STATUS assigns 'STATUS to check and act on buttons and the reference analog channels mentioned above.

1239

```
0 \ Plotting dimensions
1 2VARIABLE BOUNDARIES
2 0 4094 BOUNDARIES 2!
3
4 10 60 ORIGIN
5 -165 550 SIZE
6 0 70 Y-RANGE
7 BOUNDARIES 2@ X-RANGE
8 2 THICKNESS !
9 4 SPACING !
10
11
12
13
14
15
```

Sets parameters for display plots.

1240

```
0 \ Plotting tools
1 : *AT ( y x)   SCALE AT ;
2 : +*DRAW ( dy dx)   >R  Y FRAME @  span @ */
3    R>  X FRAME @  span @ */  +DRAW ;
4 : CLIP ( n - n)   LOW @  DUP >R  MAX  R> span @ +  MIN ;
5 : .X   CP 2@  THICKNESS @ >R  2DUP  R@ 2/  DUP V-
6    2OVER  R@ 2/  DUP V+  RULE  R@ NEGATE  0 +AT
7    CP 2@  R>  DUP NEGATE V+  LINE  CP 2! ;
8 : X'S   BRUSH -1 -SP C!  'BRUSH ASSIGN ( y x)
9    CP 2@ V+ AT  .X ;
10 : o'S   BRUSH  -1 -SP C!  'BRUSH ASSIGN ( y x)   CP 2@ V+ AT
11    THICKNESS @ 1- CIRCLE ;
12
13
14
15
```

*AT world coordinate positioning.
+*DRAW world coordinates relative drawing.
CLIP prevents excursions beyond the plotting area.
X'S and o'S assign different line types for drawing.

1241

```
0 \ Labelling
1 CREATE CRAYONS  ' PLUSES ,  ' X'S ,  ' o'S ,  ' THIN ,
2 : CRAYON ( n)   CRAYONS SWAP OF  @EXECUTE ;
3 : YLABEL   80 40 AT  DOWN G" Time"
4    3 -18 +AT  ACROSS G" (sec)"  13 -18 +AT  25 PLACE ;
5 : XLABEL  2 160 AT  45 PLACE  2 480 AT  43 PLACE
6    0 20 +AT  26 PLACE ;
7 : legend ( y x n)   >R AT R>  G" Gage # "
8    DUP 1+ 48 + PLACE  ( G"  ")  CRAYON  3 0 +AT  0 50 +DRAW ;
9 : LEGEND   180 40  2DUP 0 legend  2DUP 160  DUP >R  + 1 legend
10    SWAP 10 + SWAP  2DUP 2 legend  R> + 3 legend ;
11
12
13
14
15
```

CRAYON selects from the line types stored in CRAYONS.
YLABEL labels the vertical axis.
XLABEL labels the horizontal axis.
LEGEND shows the line type for each gage.

1242

```
 0 \ Plot setup
 1 30
 2 : NOTCH ( n)   TRIGGER SWAP  2DUP LOWER @
 3    0 SWAP [ DUP ] LITERAL  OVER  OFF *DRAW *DRAW
 4    UPPER @ 0 SWAP LITERAL OVER  OFF *DRAW *DRAW ;
 5 : NOTCHES   4 0 DO  I CRAYON  I NOTCH  LOOP ;
 6
 7 - : SHEET   HR THIN   CORNER 2@  2DUP AT
 8    0 X FRAME @ +DRAW   Y FRAME @ 0  2DUP +DRAW  0 ZERO *AT
 9    2DUP +DRAW  2SWAP AT +DRAW
10    CORNER 2@   0 ZERO SCALE ORIGIN Y-AXIS  ORIGIN ;
11
12
13
14
15
```

SHEET displays an empty plotting sheet with boundaries.
NOTCHES marks trigger values for each gage.

1243

```
 0 \ Stress vs time
 1 2VARIABLE TALARM0    VARIABLE gage
 2 : >GAGE ( n)   2* 4 +  gage ! ;
 3 : GAGE ( a - a)   gage @ + ;
 4 : TREL ( d - n)   TALARM0 2@  D- DROP
 5    [ T/MS 1000 / SWAP ] 2LITERAL  */ROUND ;
 6 - : PLOT ( a n)   >GAGE  OFF   ENTRIES OVER + SWAP
 7    DO I 2@ TREL Y CLIP   I GAGE @ X CLIP  *DRAW
 8    SAMPLE +LOOP ;
 9 : PLOTS ( a)    DUP 2@ TALARM0 2!   4 0 DO PAUSE?
10    I CRAYON  DUP I PLOT  LOOP  DROP ;
11 - : DISPLAY   SHEET  XLABEL YLABEL LEGEND
12    NOTCHES  REPLAY PLOTS   LIGHT off ;
13
14
15
```

TALARM0 holds the start time of an alarm event.
TREL is the relative time from the start of an event.
gage holds the offset from start of sample for the current gage
>GAGE and GAGE are used to set and use 'gage'.

PLOTS plots a different line type for each gage for the event
       at REPLAY.
DISPLAY sets up a plotting sheet then calls PLOTS.
         Stress is horizontal with time vertical.

1244

1245

```
0 \ Display aids
1 VARIABLE 'DATE    VARIABLE 'alive
2 : */ROUND ( n n n - n)   DUP >R  */MOD  SWAP 2*
3    R> < NOT  IF 1+ THEN ;
4 : .date ( n)   Y-MD MD  0 <# # # 2DROP  47 HOLD
5    0 # # 2DROP 47 HOLD  0 # # #>  TYPE SPACE ;
6 : .READ-ERR   .ERR ." Error reading disk" ;
7 : .alive   'alive @  0=  DUP 'alive !  2+ EMIT ;
8 : .ALIVE   C# 2@   1 40 TAB  .alive  TAB ;
9 : .PAGE ( a - f)   (D  1  20 0 DO  OVER EXECUTE  0=
10    IF  1- LEAVE THEN  LOOP  D)  SWAP DROP ;
11
12
13
14
15
```

*/ROUND behaves as */ with rounding.
'DATE holds the initial date of the current log file.
.date converts and prints a number into a date of form YYMMDD.
.PAGE writes a screenful of data using the address of the
one-line-of-output word stored on the stack.
Output flag set if output word's flag is reset.

1246

```
0 \ Banner
1 : .SINCE   0. 0 handle @  -SEEK  IF  .READ-ERR CR
2    ELSE  PORTAL  24  handle @  -GETS  IF  .READ-ERR CR
3    ELSE  ." Disk installed at "  PORTAL 4 + DUP
4    @ DUP  'DATE ! .date   2+ 2@ .TIME   THEN THEN ;
5 : %LEFT ( - n)   0 DRIVE?  100 SWAP */ROUND  >R 2DROP R> ;
6 : .USAGE   C# 2@  0 49 TAB  ." Events="  #EVENTS @ 5 U.R
7    3 SPACES  ." Space left="  %LEFT 4 U.R ." %"  TAB ;
8 : BANNER   .SINCE .USAGE  2 0 !TOP  24 0 TAB ;
9 : monochrome   MODE @  3 > IF  TX BANNER  THEN ;
10 : .ALARM   C# 2@  0 32 TAB " _____ALARM_____"
11    COUNT MARK TAB ;
12
13
14
15
```

BANNER displays a banner showing the file start date/time
number of events, and % of disk space remaining.
A window is setup to hold these values without scrolling
monochrome if not already in text mode,
reconfigures screen to text mode and displays BANNER.
.ALARM prints an alarm message in the banner area.

1247

```
0 \ Event summary
1 : +sample ( a h - n f)   >R DUP SAMPLE R>  -gets
2    IF  .READ-ERR CR  ABORT  THEN   SWAP @  0< ;
3 : +EVENT ( - n)   handle @ >R  BEGIN  PAUSE?  PORTAL R@ +sample
4    OVER AND  WHILE   PORTAL 2+ @ 'DATE !   SAMPLE 0 1 R@ -SEEK
5    IF  .READ-ERR CR  ABORT  THEN   DROP  REPEAT
6    DUP IF  CR ." Event at "  'DATE @ .date  PORTAL 2@ .TIME
7    [ ENTRIES SAMPLE - 0 ] 2LITERAL  1 handle @ -SEEK
8    IF  .READ-ERR CR  ABORT  THEN   THEN R> DROP ;
9 : EVENTS   monochrome  PAGE BANNER
10    BEGIN   ['] +EVENT .PAGE   WHILE  CR ." More"  Y/N   NOT
11    IF  CR EXIT  THEN   REPEAT
12    5 SPACES ." End of file reached " ;
13
14
15
```

+sample reads one sample worth of data. Returns count & flag.
The flag is true if the sample is a regular record.
+EVENT seacrhes the file for the next event record.
Regular records are skipped.
EVENTFUL prints a page of events.
EVENTS prints events a page at a time.

1248

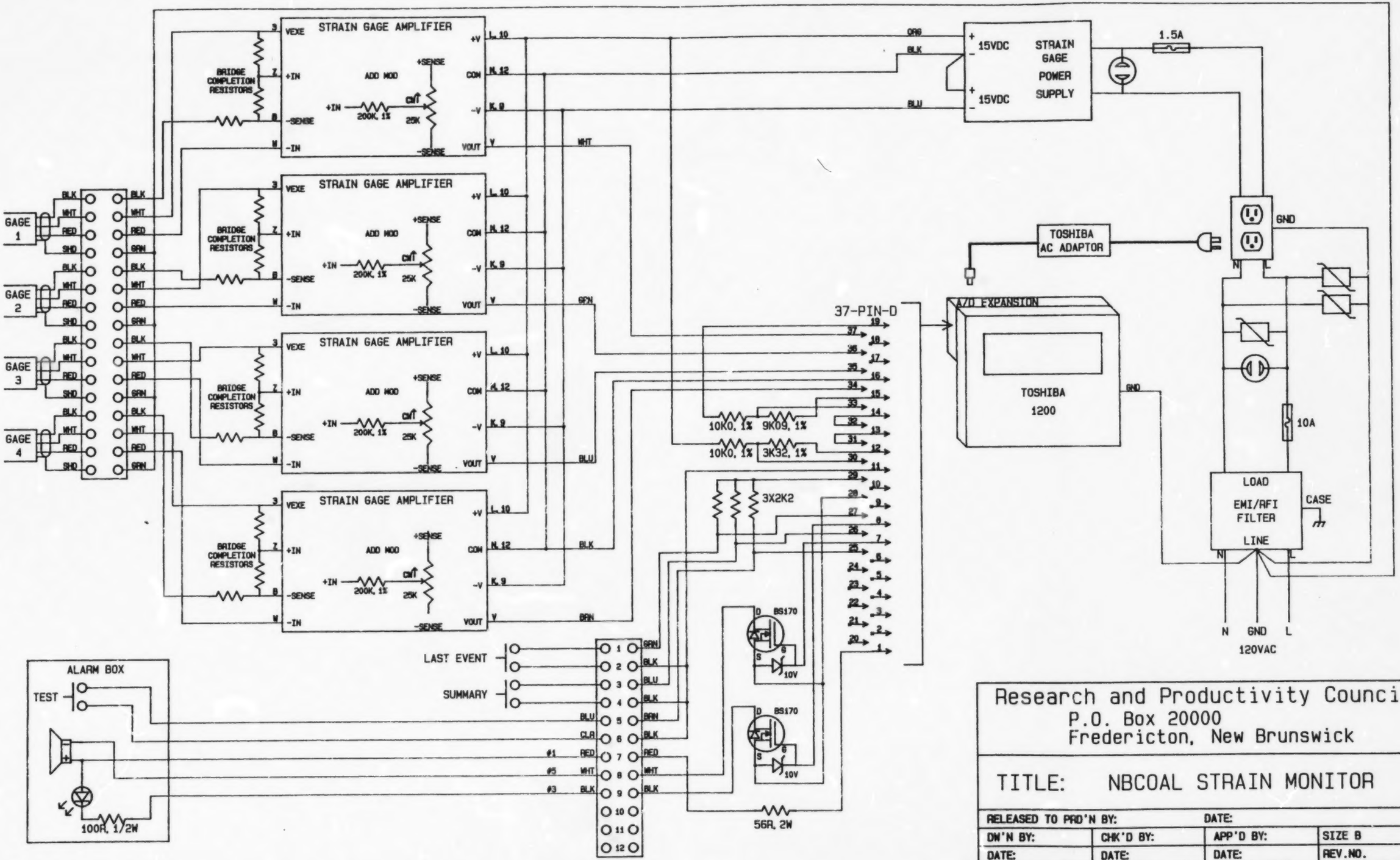
```
0 \ Routine records summary
1 VARIABLE 'stress        : stress   'stress @EXECUTE ;
2 : INTERNAL   'stress ASSIGN ( n n - n)   DROP ;   INTERNAL
3 : EXTERNAL   'stress ASSIGN ( n n - n)   STRESS ;
4 : .REGULAR ( a)   CR 2+  DUP @ .date  2+ DUP 2@ .TIME 4 +
5    5 1 DO  2 SPACES  I 1 U.R ." :"  DUP I 1- 2* OF
6    DUP @ I stress S>D  6 D.R   2+ @ I stress S>D  6 D.R
7    LOOP DROP ;
8 : +REGULAR ( - n)   handle @ >R  BEGIN  PAUSE
9    PORTAL R@ +sample  NOT OVER AND   WHILE  DROP  REPEAT
10   DUP IF  PORTAL SAMPLE +  R@ +sample 2DROP
11   PORTAL .REGULAR  THEN  R> DROP ;
12 : ROUTINE   24. 0 handle @ -SEEK DROP
13   BEGIN   ['] +REGULAR .PAGE   WHILE  CR ." More"  Y/N  NOT
14   IF  CR EXIT  THEN   REPEAT
15   5 SPACES ." End of file reached " ;
```

```
.REGULAR  writes one line of a regular record.
          (date, time, min & max for each gage)
+REGULAR  finds and reads next regular record.
          Returns number of bytes of last read.
ROUTINE   displays all regular records.
```

1249

1250

Appendix D. Monitor Wiring Diagram



Research and Productivity Council
P.O. Box 20000
Fredericton, New Brunswick

TITLE: NBCOAL STRAIN MONITOR

RELEASED TO PRD'N BY:		DATE:	
DW'N BY:	CHK'D BY:	APP'D BY:	SIZE B
DATE:	DATE:	DATE:	REV.NO.